# Solution sets for equations over free groups are EDT0L languages — ICALP 2015 version*

Laura Ciobanu[1], Volker Diekert[2], and Murray Elder[3]

[1] Institut de mathématiques, Université de Neuchâtel, Switzerland
[2] Institut für Formale Methoden der Informatik, Universität Stuttgart, Germany
[3] School of Mathematical & Physical Sciences, The University of Newcastle, Australia

**Abstract** We show that, given a word equation over a finitely generated free group, the set of all solutions in reduced words forms an EDT0L language. In particular, it is an indexed language in the sense of Aho. The question of whether a description of solution sets in reduced words as an indexed language is possible has been been open for some years [9, 10], apparently without much hope that a positive answer could hold. Nevertheless, our answer goes far beyond: they are EDT0L, which is a proper subclass of indexed languages. We can additionally handle the existential theory of equations with rational constraints in free products $\star_{1\le i\le s} F_i$, where each $F_i$ is either a free or finite group, or a free monoid with involution. In all cases the result is the same: the set of all solutions in reduced words is EDT0L. This was known only for quadratic word equations by [8], which is a very restricted case. Our general result became possible due to the recent recompression technique of Jeż. In this paper we use a new method to integrate solutions of linear Diophantine equations into the process and obtain more general results than in the related paper [5]. For example, we improve the complexity from quadratic nondeterministic space in [5] to quasi-linear nondeterministic space here. This implies an improved complexity for deciding the existential theory of non-abelian free groups: $\mathsf{NSPACE}(n \log n)$. The conjectured complexity is $\mathsf{NP}$, however, we believe that our results are optimal with respect to space complexity, independent of the conjectured $\mathsf{NP}$.

## Introduction and main results

The first algorithmic description of all solutions to a given equation over a free group is due to Razborov [17, 18]. His description became known as a *Makanin-Razborov diagram*. This concept plays a major role in the positive solution of Tarski's conjectures about the elementary theory in free groups [12, 21].

It was however unknown that there is an amazingly simple formal language description for the set of all solutions of an equation over free groups in reduced words: they are EDT0L. An EDT0L language $L$ is given by a nondeterministic

* Research supported by the Australian Research Council FT110100178 and the University of Newcastle G1301377. The first author was supported by a Swiss National Science Foundation Professorship FN PP00P2-144681/1. The first and third authors were supported by a University of Neuchâtel Overhead grant in 2013.

finite automaton (NFA), where transitions are labeled by endomorphisms in a free monoid which contains a symbol #. Such an NFA defines a rational language $\mathcal{R}$ of endomorphisms, and the condition on $L$ is that $L = \{h(\#) \mid h \in \mathcal{R}\}$. The NFA we need for our result can be computed effectively in nondeterministic quasi-linear space, i.e., by some $\mathsf{NSPACE}(n \log n)$ algorithm. As a consequence, the automaton has singly exponential size $2^{\mathcal{O}(n \log n)}$ in the input size $n$.

A description of solution sets as EDT0L languages was known before only for quadratic word equations by [8]; the recent paper [5] did not aim at giving such a structural result. There is also a description of all solutions for a word equation by Plandowski in [14]. His description is given by some graph which can computed in singly exponential time, but without the aim to give any formal language characterization. Plandowski claimed in [14] that his method applies also to free groups with rational constraints, but he found a gap [15].

The technical results are as follows. Let $\mathrm{F}(A_+)$ be the free group over a finite generating set $A_+$ of (positive) letters. We let $A_\pm = A_+ \cup \left\{a^{-1} \;\middle|\; a \in A_+\right\} \subseteq \mathrm{F}(A_+)$. We view $A_\pm$ as a finite alphabet (of *constants*) with the involution $\overline{a} = a^{-1}$. The involution is extended to the free monoid $A_\pm^*$ by $\overline{a_1 \cdots a_k} = \overline{a_k} \cdots \overline{a_1}$. We let $\pi : A_\pm^* \to \mathrm{F}(A_+)$ be the canonical morphism. As a set, we identify $\mathrm{F}(A_+)$ with the rational (i.e., regular) subset of reduced words inside $A_\pm^*$. A word is *reduced* if it does not contain any factor $a\overline{a}$ where $a \in A_\pm$. Thus, $w \in A_\pm^*$ is reduced if and only if $\pi(w) = w$. We emphasize that $\mathrm{F}(A_+)$ is realized as a subset of $A_\pm^*$. Let $\Omega$ be a set of *variables* with involution. An *equation* over $\mathrm{F}(A_+)$ is given as a pair $(U, V)$, where $U, V \in (A_\pm \cup \Omega)^*$ are words over constants and variables. A *solution* of $(U, V)$ is a mapping $\sigma : \Omega \to A_\pm^*$ which respects the involution such that $\pi\sigma(U) = \pi\sigma(V)$ holds in $\mathrm{F}(A_+)$. As usual, $\sigma$ is extended to a morphism $\sigma : (A_\pm \cup \Omega)^* \to A_\pm^*$ by leaving constants invariant. Throughout we let # denote a special symbol, whose main purpose is to encode a tuple of words $(w_1, \ldots, w_k)$ as a single word $w_1 \# \cdots \# w_k$.

**Theorem 1.** *Let $(U, V)$ be an equation over $F(A_+)$ and $\{X_1, \ldots, X_k\}$ be any specified subset of variables. Then the solution set* $\mathrm{Sol}(U, V)$ *is EDT0L where* $\mathrm{Sol}(U, V) = \{\sigma(X_1)\# \cdots \#\sigma(X_k) \mid \sigma$ *solves* $(U, V)$ *in reduced words*$\}$.

*Moreover, there is a nondeterministic algorithm which takes $(U, V)$ as input and computes an NFA $\mathcal{A}$ such that* $\mathrm{Sol}(U, V) = \{\varphi(\#) \mid \varphi \in L(\mathcal{A})\}$ *in quasi-linear space.*

The statement of Theorem 1 shifts the perspective on how to solve equations. Instead of solving an equation, we focus on an effective construction of some NFA producing the EDT0L set. Once the NFA is constructed, the existence of a solution, or whether the number of solutions is zero, finite or infinite, become graph properties of the NFA.

Theorem 1 is a special case of a more general result involving the existential theory with rational constraints over free products. The generalization is done in several directions. First, we can replace $\mathrm{F}(A_+)$ by any finitely generated free product $\mathbb{F} = \star_{1 \le i \le s} F_i$ where each $F_i$ is either a free or finite group, or a free monoid with arbitrary involutions (including the identity). Thus, for example

we may have $\mathbb{F} = \{a,b\}^* \star \mathbb{Z} \star \mathrm{PSL}(2,\mathbb{Z}) = \{a,b\}^* \star \mathbb{Z} \star (\mathbb{Z}/3\mathbb{Z}) \star (\mathbb{Z}/2\mathbb{Z})$ where $\overline{a} = a$ and $\overline{b} = b$. Second, we allow arbitrary rational constraints. We consider Boolean formulae $\Phi$, where each atomic formula is either an equation or a rational constraint, written as $X \in L$, where $L \subseteq \mathbb{F}$ is a rational subset.

Allowing rational constraints makes it necessary to specify how the input for a constraint is given. We do so algebraically, by using a morphism $\rho : A^* \to N$, where $N$ is a finite monoid with involution and $A = A^{-1} \subseteq \mathbb{F}$ generates $\mathbb{F}$. Thus, we write a constraint in the form $X \in \rho^{-1}(m)$, with $m \in N$, and the interpretation $\rho\sigma(X) = m$. The input size $\|\Phi\|$ is given by the sum of the lengths of all atomic formulae, together with $(|A| + |\Omega|)(1 + \log |N|)$. The specification that the solution is in reduced words increases the input size by at most a factor of $\mathcal{O}(\log |A|)$.

**Theorem 2.** *Let $\mathbb{F}$ be a free product as above, $\Phi$ a Boolean formula over equations and rational constraints, and $\{X_1, \ldots, X_k\}$ any subset of variables. Then $\mathrm{Sol}(\Phi) = \{\sigma(X_1)\# \cdots \#\sigma(X_k) \mid \sigma$ solves $\Phi$ in reduced words$\}$ is EDT0L.*

*Moreover, there is an algorithm which takes $\Phi$ as input and produces an NFA $\mathcal{A}$ such that $\mathrm{Sol}(\Phi) = \{\varphi(\#) \mid \varphi \in L(\mathcal{A})\}$. The algorithm is nondeterministic and uses quasi-linear space in the input size $\|\Phi\|$.*

The proof of Theorem 1 is given in Section 2. Part II of the paper is devoted to the proof of Theorem 2 which is more technical and more difficult.

## 1 Preliminaries

We assume that the reader is familiar with big-$\mathcal{O}$ and big-$\Theta$ notation. A function $f : \mathbb{N} \to \mathbb{R}$ is called *quasi-linear* if we have $|f(n)| \in \mathcal{O}(n \ \log n)$. For results and notation in complexity theory we refer to the textbook [13]. We also use standard notation from combinatorics on words and automata theory according to [7].

### 1.1 Words and involutions

If $A$ is a set then $A^*$ denotes the *free monoid over* $A$. An element of $A$ is called *letter* and an element of $A^*$ is called *word*. The length of word $w$ is denoted by $|w|$, and $|w|_a$ counts how often a letter $a$ appears in $w$.

If $M$ is any monoid and $u, v \in M$, then we write $u \leq v$ if $u$ is a *factor* of $v$, which means we can factorize $v = xuy$ for some $x, y \in M$. We denote the neutral element in $M$ by 1. In particular, 1 denotes also the empty word.

An *involution* of a set $A$ is a mapping $x \mapsto \overline{x}$ such that $\overline{\overline{x}} = x$ for all $x \in A$. For example, the identity map is an involution. A *morphism* between sets with involution is a mapping respecting the involution. A *monoid with involution* has to additionally satisfy $\overline{xy} = \overline{y}\,\overline{x}$. A *morphism* between monoids with involution is a homomorphism $\varphi : M \to N$ such that $\varphi(\overline{x}) = \overline{\varphi(x)}$. It is an *$S$-morphism* if $\varphi(x) = x$ for all $x \in S \subseteq M$. All groups are monoids with involution given by $\overline{x} = x^{-1}$, and all group-homomorphisms are morphisms. Any involution on a set $A$ extends to $A^*$: for a word $w = a_1 \cdots a_m$ we let $\overline{w} = \overline{a_m} \cdots \overline{a_1}$. If $\overline{a} = a$ for all

$a \in A$ then $\overline{w}$ is simply the word $w$ read from right-to-left. The monoid $A^*$ is called a *free monoid with involution*.

### 1.2 NFAs, rational and recognizable subsets in monoids

Let us recall the notion of recognizable and rational set and let us emphasize that this concerns incomparable families, in general. See [7] for more background. The term *regular* will be used only in the context of finitely generated free monoids.

Let $M$ be any monoid. A subset $L \in M$ is called *recognizable* if there is a homomorphism $\psi : M \to N$ to some finite monoid $N$ such that $L = \psi^{-1}(\psi(L))$. We also say that $N$ or $\psi$ *recognizes* $L$. Recognizability is a "saturation property": we have $w \in L$ if and only if $\psi(w) \in \psi(L)$.

The family of *rational subsets* $\mathrm{RAT}(M)$ is defined inductively as follows. All finite subsets of $M$ are rational. If $L, L' \subseteq M$ are rational, then the union $L \cup L'$, the concatenation $L \cdot L'$, and $L^+$ are rational. Define $L^0 = \{1\}$ and $L^{i+1} = L \cdot L^i$ for $i \in \mathbb{N}$. Then $L^+ = \bigcup \{L^i \mid i > 0\}$ denotes the subsemigroup of $M$ which is generated by the subset $L \subseteq M$. We let $L^* = L^+ \cup \{1\}$; then $L^*$ is the corresponding submonoid.

For a finitely generated free monoid $A^*$ the family $\mathrm{RAT}(A^*)$ coincides with the family of recognizable subsets: this is the content of Kleene's classical theorem (see any standard textbook on formal languages, such as [7]). In general, however, the two families are incomparable. For example, for a group $G$ the two families coincides if and only if $G$ is finite.

By definition, if $h : M \to M'$ is a homomorphism, then $L \mapsto h(L)$ induces a mapping $\mathrm{RAT}(M) \to \mathrm{RAT}(M')$. The mapping is surjective if and only if the homomorphism $h$ is surjective. In the following let $M$ be finitely generated. Consider any surjective homomorphism $\pi : \Gamma^* \to M$ where $\Gamma$ is finite. Then every $L \in \mathrm{RAT}(M)$ can be specified by some $K \in \mathrm{RAT}(\Gamma^*)$ such that $\pi(K) = L$. The family $\mathrm{RAT}(\Gamma^*)$ coincides with the family of "regular" subsets of $\Gamma^*$. Regular subsets of $\Gamma^*$ are those which can be accepted by a non-deterministic finite automaton, i.e. an NFA. Again, we can define the notion of NFA for arbitrary monoids $M$: an NFA $\mathcal{A}$ over $M$ is a tuple $\mathcal{A} = (Q, M, \delta, I, F)$ where $Q$ is a set of *states*, $I \subseteq Q$ is the set of *initial* states, $F \subseteq Q$ is the set of *final* states, and $\delta \subseteq Q \times M \times Q$ is a finite set of *transitions*. If $(p, m, q)$ is a transition then $m \in M$ is called its *label* and each path

$$(p_0, m_1, p_1), (p_1, m_2, p_2), \ldots, (p_{k-1}, m_k, p_k)$$

labels a monoid element $m_1 \cdots m_k \in M$. For states $p, q \in Q$ we let $L(\mathcal{A}, p, q) \subseteq M$ be the set of labels of paths from $p$ to $q$. Thus, the *accepted language* $L(\mathcal{A})$ of an NFA $\mathcal{A}$ is

$$L(\mathcal{A}) = \bigcup \{L(\mathcal{A}, p, q) \mid p \in I \wedge q \in F\}.$$

Moreover, if a regular set $K \subseteq \Gamma^*$ is accepted by some NFA $\mathcal{A}$ with state set $Q = \{1, \ldots, n\}$, then we can choose for a recognizing homomorphism $N$ the

monoid of Boolean $n \times n$ matrices $\mathbb{B}^{n\times n}$. Indeed, for each letter $a \in \Gamma$ define a matrix $\rho(a) \in \mathbb{B}^{n\times n}$ by

$$\rho(a)_{ij} = \begin{cases} 1 & \text{if } a \in L(\mathcal{A}, i, j) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Cleary, due to (1) we have for all $w \in \Gamma^*$ the equivalence $w \in L(\mathcal{A}) \subseteq \Gamma^* \iff \rho(w) \in \rho(L(\mathcal{A})) \subseteq \mathbb{B}^{n\times n}$.

We say that a finite monoid (with involution) $N$ has an *efficient* representation if we can specify each element $m \in N$ by $\mathcal{O}(\log |N|)$ bits and if all basic operations, like equality checking, (computing the involution), and multiplication, are in space $\mathcal{O}(\log |N|)$, too. For example, $\mathbb{B}^{n\times n}$ has an efficient representation (if the involution is the transposition).

In the following, all finite monoids used to define rational constraints are assumed to have an efficient representation. Throughout we use the following fact: if $\psi_1 : M \to N_1$ recognizes $L_1 \subseteq M$ and $\psi_2 : M \to N_2$ recognizes $L_2 \subseteq M$, then $\psi_1 \times \psi_2 : M \to N_1 \times N_2$ recognizes every Boolean combination of $L_1$ and $L_2$. Thus, whenever we add a new constraint by another recognizing homomorphism we switch to a larger monoid. There is, however, no size explosion because we use finite monoids with an efficient representation and $\log |N_1 \times N_2| = \log |N_1| + \log |N_2|$.

### 1.3 EDT0L systems

The notion of *EDT0L system* refers to **E***xtended,* **D***eterministic,* **T***able,* **0** *interaction, and* **L***indenmayer*. There is a vast literature on Lindenmayer systems, see [19], with various acronyms such as D0L, DT0L, ET0L, etc. The subclass EDT0L is equal to HDT0L (see e.g. [20, Thm. 2.6]), and has received particular attention. We use very little L-theory: essentially we content ourselves to define EDT0L through a characterization (using rational control) due to Asveld [2]. The class of EDT0L languages is a proper subclass of indexed languages in the sense of [1], see [6]. For more background we refer to [20].

**Definition 1.** *Let $A$ be an alphabet and $L \subseteq A^*$ be a subset. We say that $L$ is* EDT0L *if there is an alphabet $C$ with $A \subseteq C$, a finite set $H \subseteq \mathrm{End}(C^*)$ of endomorphisms of $C$, a rational language $R \subseteq H^*$, and a symbol $\# \in C$ such that $L = \{\varphi(\#) \mid \varphi \in R\}$.*

Note that for a subset $R \subseteq H^*$ of endomorphisms of $C^*$ we have $\{\varphi(\#) \mid \varphi \in R\}$ is a subset of $C^*$. Our definition implies that $R$ must guarantee that $\varphi(\#) \in A^*$ for all $\varphi \in R$. The language $C$ is called an *extended alphabet*.

*Example 1.* Let $A = \{a, b\}$ and $C = \{a, b, \#, \$\}$. We let $H$ be set of four endomorphisms $f, g_a, g_b, h$ satisfing $f(\#) = \$\$$, $g_a(\$) = \$a$, $g_b(\$) = \$b$, and $h(\$) = 1$, and on all other letters the $f, g_a, g_b, h$ behave like the identity. Consider the rational language $R \subseteq H^*$ defined by $R = h\{g_a, g_b\}^* f$ (where endomorphisms are applied right-to-left). A simple inspection shows that $\{\varphi(\#) \mid \varphi \in R\} = \{vv \mid v \in A^*\}$, which is not context-free.

### 1.4 Triangulation

We can replace an equation over any monoid $M$ by a system of *triangular equations* (i.e., equations $X = V$ where $|V| = 2$) and one additional special equation $Y = 1$, where $Y$ is a fresh variable. The procedure is straightforward: consider an equation $U = V$ where $U, V \in (A \cup \Omega)^*$ are words, $A \subseteq M$, and $\Omega = \{X_1 \dots X_k\}$ is a set of (free) variables. Clearly,

$$\forall X_1, \dots, X_k : (U = V \iff \exists X, Y : \ X = UYY \wedge X = VYY \wedge Y = 1).$$

With the exception $Y = 1$, both equations have the form $X = W$ with $|W| \geq 2$. If $|W| \geq 2$ then $W = yzW'$ and we obtain

$$\forall X, Y, X_1, \dots, X_k : X = W \iff \exists X' : \ (X' = yz) \wedge (X = X'W').$$

Using this we can transform any system of equations into a new system with additional variables; and we end up with a system of equations of type $X = yz$ and one (singular) equation $Y = 1$.

## 2 Proof of Theorem 1

### 2.1 Preprocessing.

According to Section 1.4 we start with a system of triangular of equations: there is one special equation $Y = 1$ and all other equations have the form $X = V$ where $X$ is a variable and $|V| = 2$. Next, we add the special symbol $\#$ to the alphabet $A_\pm$; and we define $A = A_\pm \cup \{\#\}$. We let $\overline{\#} = \#$; this will be the only self-involuting letter in the proof of Theorem 1. We must make sure that no solution uses $\#$ and every solution is in reduced words. We do so by introducing a finite monoid $N$ with involution which plays the role of (a specific) rational constraint.

Let $N = \{1, 0\} \cup A_\pm \times A_\pm$ have multiplication $1 \cdot x = x \cdot 1 = x$, $0 \cdot x = x \cdot 0 = 0$, and

$$(a, b) \cdot (c, d) = \begin{cases} 0 & \text{if } b = \overline{c} \\ (a, d) & b \neq \overline{c}. \end{cases} \tag{2}$$

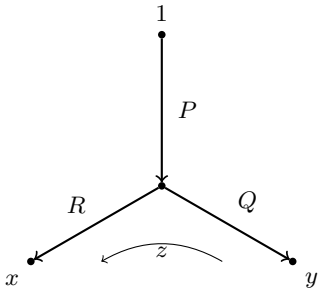
The monoid $N$ has an involution by $\overline{1} = 1$, $\overline{0} = 0$, and $\overline{(a, b)} = (\overline{b}, \overline{a})$. Consider the morphism $\mu_0 : A^* \to N$ given by $\mu_0(\#) = 0$ and $\mu_0(a) = (a, a)$ for $a \in A_\pm$. It is clear that $\mu_0$ respects the involution and $\mu_0(w) = 0$ if and only if either $w$ contains $\#$ or $w$ is not reduced. If, on the other hand, $1 \neq w \in A_\pm^*$ is reduced, then $\mu_0(w) = (a, b)$, where $a$ is the first and $b$ the last letter of $w$. Also, $\mu_0(w) = 1$ if and only if $w = 1$. Thus, if $\sigma$ is a solution in reduced words, then for each variable $X$ there exists some $0 \neq \mu(X) = \overline{\mu(\overline{X})} \in N$ with $\mu(X) = \mu_0\sigma(X)$.

The morphism $\mu$ allows us to remove the special equation $Y = 1$. It is replaced by fixing $\mu(Y) = 1$.

For the other variables there are only finitely many choices for $\mu$, so we assume that each equation is specified together with a morphism $\mu : \Omega \longrightarrow N$. We now

require that a solution $\sigma : \Omega \to A^*$ satisfy three properties: $\pi\sigma(U) = \pi\sigma(V)$, $\overline{\sigma(X)} = \sigma(\overline{X})$, and $\mu(X) = \mu_0\sigma(X)$ for all $X \in \Omega$.

The next step allows us to work with free monoids with involution rather than with groups. This relies on Lemma 1, which is well-known, too. Its geometric interpretation is simply that the Cayley graph of a free group (over standard generators) is a tree. See Figure 1 for a visual explanation. □




$$\forall x, y, z\ \exists P, Q, R:\ \pi(x) = \pi(yz) \iff \pi(x) = PR \wedge \pi(y) = PQ \wedge \pi(z) = \overline{Q}R. \quad (3)$$

**Figure1.** Part of the Cayley graph of $\mathrm{F}(A_+)$ with standard generators: the root is 1 on the top. The geodesics to vertices $x$ and $y$ split after an initial path labeled by $P$.

**Lemma 1.** *Let $x, y, z$ be reduced words in $A_\pm^*$. Then $x = yz$ in the group $F(A_+)$ if and only if there are reduced words $P, Q, R$ in $A_\pm^*$ such that $x = PR$, $y = PQ$, and $z = \overline{Q}R$ holds in the free monoid $A_\pm^*$.*

*Proof.* If there are words $P, Q, R$ in $A_\pm^*$ such that $x = PR$, $y = PQ$, and $z = \overline{Q}R$ holds in the free monoid $A_\pm^*$ then we have $\pi(x) = \pi(yz)$, hence $x = yz$ in the group $\mathrm{F}(A_+)$. This is trivial and holds whether or not $x, y, z$ are reduced. For the other direction, let $x, y, z$ be reduced words in $A_\pm^*$ such that $\pi(x) = \pi(yz)$. If the word $yz$ is reduced then we choose $P = y$, $Q = 1$, and $R = z$. In the other case we have $y = y'a$ and $z = \overline{a}z'$ for some $a \in A_\pm$ and we can use induction. □

The consequence of Lemma 1 is that with the help of fresh variables $P, Q, R$ we can substitute every equation $x = yz$ with $x, y, z \in A_\pm \cup \Omega$ by the following three word equations to be solved over a free monoid with involution.

$$x = PR, \qquad y = PQ, \qquad z = \overline{Q}R. \quad (4)$$

The enlarged set of variables becomes $\Omega \cup \{P, \overline{P}, Q, \overline{Q}, R, \overline{R}\}$. Assume $\sigma$ solves (4) such that $\sigma(x)$, $\sigma(y)$, $\sigma(z)$ are reduced. Let $\sigma(P) = p$, $\sigma(Q) = q$, and $\sigma(R) =$

$r$, Then $p$, $q$, and $r$ are reduced and there is no cancellation in any of the words $pr$, $pq$, and $\overline{q}r$. The lack of cancellation is encoded by $\mu$, but note that there are various choices for $\mu(P), \mu(Q), \mu(R)$: for example, let $\mu(x) = (b, c)$, $\mu(y) = (b, a)$, and $\mu(z) = (\overline{a}, c)$ and assume that $p, q, r \neq 1$. Then there are a last letter $d$ in $p$ and first letters $e$ in $q$ and $f$ in $r$ such that $\overline{d} \neq e \neq f \neq \overline{d}$, so the choice $\mu(P) = (b, d)$, $\mu(Q) = (e, a)$, and $\mu(R) = (f, c)$ is correct.

### 2.2 The initial equation

By Lemma 1 it is enough to prove the analogue of Theorem 1 for free monoids with involution, systems of equations $(U_i, V_i)$, $1 \leq i \leq s$ and a morphism $\mu_{\text{init}} : (A \cup \Omega)^* \to N$ such that $0 \neq \mu_{\text{init}}(U_i) = \mu_{\text{init}}(V_i)$ for all $i$. Finally, we construct a single equation $(U', V')$ over $A \cup \Omega$, where $U' = U_1 \# \cdots \# U_s$ and $V' = V_1 \# \cdots \# V_s$. Notice that $\mu_{\text{init}}(X) \neq 0$ and $|U_i|_\# = |V_i|_\# = 0$ for all $i$. A solution $\sigma : \Omega \to A_\pm^*$ must satisfy $\sigma(U') = \sigma(V')$, $\overline{\sigma(X)} = \sigma(\overline{X})$, and $\mu_{\text{init}}(X) = \mu_0\sigma(X)$ for all $X \in \Omega$. The set of variables $\{X_1, \ldots, X_k\}$, specified in Theorem 1, is a subset of the new and larger set of variables $\Omega$, and the original solution set became a finite union of solution sets with respect to different choices for $\mu_0$ and $\mu_{\text{init}}$. The result holds since EDT0L is closed under finite union. In order to achieve our result we have to *protect* each variable $X_i$ as follows. We assume $A_\pm \cup \Omega = \{x_1, \ldots, x_\ell\}$ with $x_i = X_i$ for $1 \leq i \leq k$ where $\{X_1 \ldots, X_k\}$ is the specified subset in the statement of Theorem 1.

The word $W_{\text{init}}$ over $(A \cup \Omega)^*$ is then defined as:

$$W_{\text{init}} = \#x_1 \# \ldots \# x_\ell \# U' \# V' \# \overline{U'} \# \overline{V'} \# \overline{x_\ell} \# \ldots \# \overline{x_1} \#. \tag{5}$$

Observe that $W_{\text{init}}$ is longer than (but still linear in) $|A| + |\Omega| + |UV|$. The number of #'s in $W_{\text{init}}$ is odd; and if $\sigma : (A \cup \Omega)^* \to A^*$ is a morphism with $\sigma(X)$ reduced for all $X \in \Omega$, then: $\pi\sigma(U) = \pi\sigma(V) \iff \sigma(U') = \sigma(V') \iff \sigma(W_{\text{init}}) = \sigma(\overline{W_{\text{init}}})$. Here $(U, V)$ is the equation in Theorem 1 and $(U', V')$ is the intermediate word equation over $A \cup \Omega$. Therefore Theorem 1 follows by showing that the following language is EDT0L:

$$\left\{ \sigma(X_1) \# \cdots \# \sigma(X_k) \;\middle|\; \sigma(W_{\text{init}}) = \sigma(\overline{W_{\text{init}}}) \wedge \mu_{\text{init}} = \mu_0\sigma \wedge \forall X : \sigma(\overline{X}) = \overline{\sigma(X)} \right\}.$$

### 2.3 Partial commutation and extended equations.

Partial commutation is an important concept in our proof. It pops up where traditionally the unary case (solving a linear Diophantine equation) is used as a black box, as is done in [5]. At first glance it might seem like an unneccesary complication, but in fact the contrary holds. Using partial commutation allows us to encode all solutions completely in the edges of a graph, which we can construct in quasi-linear space, and is one of the major differences to [5]. As a (less important) side effect, results on linear Diophantine equations come for free as this is the special case $\mathrm{F}(A_+) = \mathbb{Z}$: solving linear Diophantine equations becomes part of a more general process.

We fix $n = n_{\text{init}} = |W_{\text{init}}|$ and some $\kappa \in \mathbb{N}$ large enough, say $k = 100$. We let $C$ be an alphabet with involution (of constants) such that $|C| = \kappa n$ and $A \subseteq C$. We define $\Sigma = C \cup \Omega$ and assume that $\#$ is the only self-involuting symbol of $\Sigma$. In the following $x, y, z, \ldots$ refer to words in $\Sigma^*$ and $X, Y, Z, \ldots$ to variables in $\Omega$. Throughout we let $B, B'$ and $\mathcal{X}, \mathcal{X}'$ denote subsets which are closed under involution and satisfy $\mathcal{X}' \subseteq \mathcal{X} \subseteq \Omega$ and either $A \subseteq B \subseteq B' \subseteq C$ or $A \subseteq B' \subseteq B \subseteq C$. In particular, $B$ and $B'$ are always comparable.

We encode a partial commutation as follows. Let $c \in B$ and either $p = c$ or $p = c\overline{c}$. (The case $p = c\overline{c}$ is needed only later in the proof of Theorem 2.) Let $\theta \subseteq (\mathcal{X} \cup B^+) \times \{p, \overline{p}\}$ denote an irreflexive and antisymmetric relation. It is called a *type* if $\theta$ satisfies the following conditions: first, $(x, y) \in \theta$ implies $(\overline{x}, \overline{y}) \in \theta$ and $x \in \mathcal{X} \cup \{a, \overline{a}, a\overline{a} \mid a \in B\} \setminus \{c, \overline{c}, c\overline{c}\}$, and second, for each $x$ we have $|\theta(x)| \leq 1$, where $\theta(x) = \{y \in B^* \mid (x, y) \in \theta\}$. The type relation $\theta$ can be stored in quasi-linear space. Given $\theta$ and $\mu : B \cup \mathcal{X} \to N$ such that $\mu(xy) = \mu(yx)$ for all $(x, y) \in \theta$, we define the following two partially commutative monoids with involution: first, $M(B, \mathcal{X}, \theta, \mu) = (B \cup \mathcal{X})^* / \{xy = yx \mid (x, y) \in \theta\}$ – a monoid with a morphism $\mu : M(B, \mathcal{X}, \theta, \mu) \to N$ – and second, $M(B, \theta, \mu) = B^* / \{xy = yx \mid (x, y) \in \theta\}$. If $w \leq W \in M(B, \mathcal{X}, \theta, \mu)$ then $w$ is called a *proper factor* if $w \neq 1$ and $|w|_{\#} = 0$.

Since the defining relations for $M(B, \mathcal{X}, \theta, \mu)$ are of the form $xy = xy$, we can define $|W|$ and $|W|_a$ for $W \in M(B, \mathcal{X}, \theta, \mu)$ by representing $W$ by some word $W \in (B \cup \mathcal{X})^*$. Typically we represent $w$, $W$ by words $w, W \in (B \cup \mathcal{X})^*$, but their interpretation is always in $M(B, \mathcal{X}, \theta, \mu)$.

**Definition 2.** *We call* $W \in M(B, \mathcal{X}, \theta, \mu)$ well-formed *if* $|W| \leq \kappa n$, $|W|_{\#} = |W_{\text{init}}|_{\#}$, *every proper factor* $x$ *of* $W$ *and every* $x \in B \cup \mathcal{X}$ *satisfies* $\mu(x) \neq 0$, *and* $B^* \cap \mu^{-1}(1) = \{1\}$. *Moreover, if* $x$ *is a proper factor then* $\overline{x}$ *is a proper factor, too. Finally, for every* $a \in A_{\pm}$ *there is a factor* $\#a\#$ *in* $W$.

**Definition 3.** *An* extended equation *is a tuple* $V = (W, B, \mathcal{X}, \theta, \mu)$ *where* $W \in M(B, \mathcal{X}, \theta, \mu)$ *is well-formed.*

*A* $B$-solution *of* $V$ *is a* $B$*-morphism* $\sigma : M(B, \mathcal{X}, \theta, \mu) \to M(B, \theta, \mu)$ *such that* $\sigma(W) = \sigma(\overline{W})$ *and* $\sigma(X) \in y^*$ *whenever* $(X, y) \in \theta$.

*A* solution *of* $V$ *is a pair* $(\alpha, \sigma)$ *such that* $\alpha : M(B, \theta, \mu) \to A^*$ *is an* $A$*-morphism satisfying* $\mu_0 \alpha = \mu$ *and* $\sigma$ *is a* $B$*-solution.*

| | |
|---|---|
| $W$ | = equation, where the solution is a "palindrome" $\sigma(W) = \overline{\sigma(W)} \in A^*$. |
| $B$ | = alphabet of constants with $\# \in A \subseteq B = \overline{B} \subseteq C$. |
| $\mathcal{X}$ | = variables appearing in $W$. Hence, $\mathcal{X} = \overline{\mathcal{X}} \subseteq \Omega$. |
| $\mu$ | = morphism to control the constraint that the solution is reduced. |
| $\theta$ | = partial commutation. |

During the process of finding a solution, we change these parameters, and we describe the process in terms of a diagram (directed graph) of states and arcs between them.

### 2.4 The directed labeled graph $\mathcal{G}$.

We are now ready to define the directed labeled graph $\mathcal{G}$ which will be the core of the NFA defining the EDT0L language $\mathrm{Sol}(U, V)$ we are aiming for.

Define the *vertex set* for $\mathcal{G}$ to be the set of all extended equations $V = (W, B, \mathcal{X}, \theta, \mu)$. The *initial vertices* are of the form $(W_{\mathrm{init}}, A, \Omega, \emptyset, \mu_{\mathrm{init}})$. Due to the different possible choices for $\mu_{\mathrm{init}}$ there are exponentially many initial vertices. In fact, Theorem 1 requires us to work in exponentially many graphs simultaneously because of the different possible $\mu_0 : A^* \to N$. However, to simplify the presentation we fix one $\mu_0$.

We define the set of *final vertices* by $\left\{(W, B, \emptyset, \emptyset, \mu) \;\middle|\; W = \overline{W}\right\}$. By definition every final vertex trivially has a $B$-solution $\sigma = \mathrm{id}_B$. (Note that in a final vertex there are no variables.) The arcs in $\mathcal{G}$ are labeled and are of the form $(W, B, \mathcal{X}, \theta, \mu) \xrightarrow{h} (W', B', \mathcal{X}', \theta', \mu')$. Here $h : C^* \to C^*$ is an endomorphism which is given by a morphism $h : B' \to B^*$ such that $h$ induces a well-defined morphism $h : M(B' \cup \mathcal{X}', \theta', \mu') \to M(B \cup \mathcal{X}, \theta, \mu)$. Note that the direction of the morphism is opposite to the direction of the arc. There are further restrictions on arcs. For example, we will have $|h(b')| \leq 3$ for all $b'$. The main idea is as follows. Suppose $(W, B, \mathcal{X}, \theta, \mu) \xrightarrow{h} (W', B', \mathcal{X}', \theta', \mu')$ is an arc, $\alpha : M(B, \theta, \mu) \to M(A, \emptyset, \mu_0)$ is an $A$-morphism, and $(W', B', \mathcal{X}', \theta', \mu')$ has a $B'$-solution $\sigma'$; then there exists a solution $(\alpha, \sigma)$ of the vertex $(W, B, \mathcal{X}, \theta, \mu)$. Moreover, for the other direction if $(\alpha, \sigma)$ solves $V = (W, B, \mathcal{X}, \theta, \mu)$ and $V$ is not final then we can follow an outgoing arc and recover $(\alpha, \sigma)$ from a solution at the target node. We will make this more precise below.

### 2.5 Compression arcs.

These arcs transform the sets of constants. Let $V = (W, B, \mathcal{X}, \theta, \mu)$ and $V' = (W', B', \mathcal{X}, \theta', \mu')$ be vertices in $\mathcal{G}$. The compression arcs have the form $V \xrightarrow{h} V'$, where either $h = \varepsilon$ is defined by the identity on $C^*$, or $h$ is defined by a mapping $c \mapsto h(c) \neq c$ with $c \in B'$. Recall that if a morphism $h$ is defined by $h(c) = u$ for some letter $c$ then, by our convention, $h(\overline{c}) = \overline{u}$ and $h(x) = x$ for all $x \in \Sigma$ which are different from $c$ and $\overline{c}$. We assume $0 \neq \mu'(c) = \mu(h(c)) \neq 1$ and $\mu(x) = \mu'(x)$ for all $x \in (B \cap B') \cup \mathcal{X}$ (if not explicitly stated otherwise).

We define compression arcs $(h(W'), B, \mathcal{X}, \theta, \mu) \xrightarrow{h} (W', B', \mathcal{X}, \theta', \mu')$ of the following types, only.

1. (**Renaming.**) We assume that $h$ is defined by $h(c) = a$ such that $B \subsetneq B' = B \cup \{c, \overline{c}\}$, and $\theta \subseteq \theta'$. Thus, possibly, $\theta \subsetneq \theta'$.
2. (**Compression.**) We have $h(c) = u$ with $|u| \leq 3$ and either $B = B'$ and $\theta' = \theta$ or $B \subsetneq B' = B \cup \{c, \overline{c}\}$ and $\theta = \theta' = \emptyset$.
3. (**Alphabet reduction.**) We have $B' \subsetneq B$, $\theta' = \emptyset$, and $h$ is induced by the inclusion $B' \subseteq B$. Hence we have $h = \varepsilon = \mathrm{id}_{C^*}$.

For the proof of Theorem 1 we compress words $u$ into a single letter $c$ only if $|u| \leq 2$. For Theorem 2 we will have in addition the case where $u = a\overline{a}c$ with

either $a = c$ (and $\overline{a} = \overline{c}$) or $(a\overline{a}, c\overline{c}) \in \theta$. The purpose of arcs of type **3** is to remove letters in $B$ that do not appear in the word $W$. This allows us to reduce the size of $B$ and to "kill" partial commutation.

**Lemma 2.** *Let $(W, B, \mathcal{X}, \theta, \mu) \xrightarrow{h} (W', B', \mathcal{X}', \theta', \mu')$ be an arc of type **1,2** or **3** with $W = h(W')$. Let $\alpha : M(B, \theta, \mu) \to M(A, \emptyset, \mu_0)$ be an $A$-morphism at vertex $V = (h(W'), B, \mathcal{X}, \theta, \mu)$ and $\sigma'$ be a $B'$-solution to $V' = (W', B', \mathcal{X}', \theta', \mu')$. Define a $B$-morphism $\sigma : M(B, \mathcal{X}, \theta, \mu) \to M(B, \theta, \mu)$ by $\sigma(X) = h\sigma'(X)$. Then $(\alpha, \sigma)$ is a solution at $V$ and $(\alpha h, \sigma')$ is a solution at $V'$ and $\alpha\sigma(W) = \alpha h\sigma'(W')$.*

*Proof.* By definition, $\mu h = \mu'$ and $\mu_0\alpha = \mu$. Hence $(\alpha h, \sigma')$ is a solution at $V$. Now, $h(X) = X$ for all $X \in \mathcal{X}$. Hence, $\sigma(h(X)) = \sigma(X) = h\sigma'(X)$. For $b' \in B'$ we obtain $\sigma h(b') = h(b') = h\sigma'(b')$ since $\sigma'$ and $\sigma$ are the identity on $B'$ and $B$ respectively. It follows $\sigma h = h\sigma'$ and $\alpha\sigma(W) = \alpha h\sigma'(W')$. Next,

$$\sigma(W) = \sigma(h(W')) = h(\sigma'(W')) = h(\sigma'(\overline{W'})) = \sigma(h(\overline{W'})) = \sigma(\overline{h(W')}) = \sigma(\overline{W}).$$

Thus, $\sigma$ is a $B$-solution to $V$ and, consequently, $(\alpha, \sigma)$ solves $V$. □

### 2.6 Substitution arcs.

These arcs transform variables. Let $V = (W, B, \mathcal{X}, \theta, \mu)$ and $V' = (W', B, \mathcal{X}', \theta', \mu')$ be vertices in $\mathcal{G}$ and $X \in \mathcal{X}$. We assume that $\mathcal{X} = \mathcal{X}' \cup \{X, \overline{X}\}$ and $\mu(x) = \mu'(x)$, as well as $\theta(x) = \theta'(x)$ for all $x \in (B \cup \mathcal{X}) \setminus \{X, \overline{X}\}$. The set of constants is the same on both sides, but $\mathcal{X}'$ might have fewer variables. Substitution arcs are defined by a morphism $\tau : \{X\} \to BX \cup \{1\}$ such that we obtain a $B$-morphism $\tau : M(B, \mathcal{X}, \theta, \mu) \to M(B, \mathcal{X}', \theta', \mu')$. We let $\varepsilon = \mathrm{id}_{C^*}$ as before. We define substitution arcs $(W, B, \mathcal{X}, \theta, \mu) \xrightarrow{\varepsilon} (\tau(W), B, \mathcal{X}', \theta', \mu')$ if one of the following conditions apply.

**4**. (**Removing a variable.**) Let $\mathcal{X}' = \mathcal{X} \setminus \{X, \overline{X}\}$. The $B$-morphism $\tau : M(B, \mathcal{X}, \theta, \mu) \to M(B, \mathcal{X}', \theta', \mu')$ is defined by $\tau(X) = 1$.

**5**. (**Variable typing.**) The purpose of this arc is to introduce some type for variables without changing anything else, so $\mathcal{X}' = \mathcal{X}$ and $\mu' = \mu$. Suppose that $\theta(X) = \emptyset$ and $p \in B^+$ is a word with $\mu(Xp) = \mu(pX)$ and such that $\theta' = \theta \cup \{(X, p), (\overline{X}, \overline{p})\}$. The $B$-morphism $\tau : M(B, \mathcal{X}, \theta, \mu) \to M(B, \mathcal{X}, \theta', \mu)$ is defined by the identity on $B \cup \mathcal{X}$. Note that the condition $\mu(Xp) = \mu(pX)$ implies that if $\mu : M(B, \mathcal{X}, \theta, \mu) \to N$ is well-defined, then $\mu : M(B, \mathcal{X}, \theta', \mu) \to N$ is well-defined, too. The other direction is trivial.

**6**. (**Substitution of a variable.**) We have $(B, \mathcal{X}, \theta) = (B', \mathcal{X}', \theta')$. Let $p \in B^+$ such that $\theta(X) \subseteq \{p\}$. (For $\theta(X) = \emptyset$ this is always true.) We suppose that we have $\mu(X) = \mu(p)\mu'(X)$ (hence, automatically $\mu(\overline{X}) = \mu'(\overline{X})\mu(\overline{p})$) and that $\tau(X) = pX$ defines a morphism $\tau : M(B, \mathcal{X}, \theta, \mu) \to M(B, \mathcal{X}, \theta, \mu')$.

**Lemma 3.** *Let $V = (W, B, \mathcal{X}, \theta, \mu) \xrightarrow{\varepsilon} (W', B, \mathcal{X}', \theta', \mu') = V'$ with $\varepsilon = \mathrm{id}_{C^*}$ be an arc of type **4,5,6** with $W' = \tau(W)$. Let $\alpha : M(B, \theta, \mu) \to M(A, \emptyset, \mu_0)$ be an $A$-morphism at vertex $V$ and $\sigma'$ be a $B$-solution to $V'$. Define a $B$-morphism*

$\sigma : M(B, \mathcal{X}, \theta, \mu) \to M(B, \theta, \mu)$ *by* $\sigma(X) = \sigma'\tau(X)$. *Then* $(\alpha, \sigma)$ *is a solution at* $V$ *and* $(\alpha, \sigma')$ *is a solution at* $V'$. *Moreover,* $\alpha\sigma(W) = \alpha h\sigma'(W')$ *where* $h = \varepsilon$ *is viewed as the identity on* $\mathrm{id}_{M(B,\theta,\mu)}$.

*Proof.* Since $\sigma'$ is a $B$-solution to $V'$ we have $\sigma(W) = \sigma'(\tau(W)) = \sigma'(\overline{\tau(W)}) = \overline{\sigma'\tau(W)} = \overline{\sigma(W)}$. Hence, $(\alpha, \sigma)$ is a solution at $V$. Since $M(B, \theta, \mu) = M(B, \theta', \mu')$ (a possible change in $\mu$ or $\theta$ concerns variables, only), $(\alpha, \sigma') =$ is a solution at $V'$. The assertion $\alpha\sigma(W) = \alpha h\sigma'(W')$ is trivial since $W' = \tau(W)$, $\sigma = \sigma'\tau$, and $h = \varepsilon$ induces the identity on $M(B, \theta, \mu)$. □

**Proposition 1.** *Let* $V_0 \xrightarrow{h_1} V_1 \cdots \xrightarrow{h_t} V_t$ *be a path in* $\mathcal{G}$ *of length* $t$, *where* $V_0 = (W_{\mathrm{init}}, A, \Omega, \emptyset, \mu_{\mathrm{init}})$ *is an initial and* $V_t = (W', B, \emptyset, \emptyset, \mu)$ *is a final vertex. Then* $V_0$ *has a solution* $(\mathrm{id}_A, \sigma)$ *with* $\sigma(W_{\mathrm{init}}) = h_1 \cdots h_t(W')$. *Moreover, we have* $W' \in \#u_1\# \cdots \#u_k\#B^*$ *such that* $|u_i|_\# = 0$ *and we can write:*

$$h_1 \cdots h_t(u_1\# \cdots \#u_k) = \sigma(X_1)\# \cdots \#\sigma(X_k), \tag{6}$$

*Proof.* By definition of final vertices we have $\overline{W'} = W'$ and no variables occur in $W'$. Hence, $\mathrm{id}_{B^*}$ defines the (unique) $B$-solution of $W'$. By definition of the arcs, $h = h_1 \cdots h_t : M(B, \emptyset, \mu) \to A^* = M(A, \emptyset, \mu_{\mathrm{init}})$ is an $A$-morphism which shows that $(h, \mathrm{id}_{B^*})$ solves $W'$. There is only one $A$-morphism at $V_0$, namely $\mathrm{id}_{A^*}$. Using Lemma 2 and Lemma 3 we see first, $V_0$ has some solution $(\mathrm{id}_{A^*}, \sigma)$ and second,

$$\mathrm{id}_{A^*}\sigma(W_{\mathrm{init}}) = \mathrm{id}_{A^*}h_1 \cdots h_t\mathrm{id}_{B^*}(W') = h_1 \cdots h_t(W'). \tag{7}$$

Finally, for $1 \le j \le t$ we have $h_j(\#) = \#$ and $|h_j(x)|_\# = 0$ for all other symbols. Hence the claim $h_1 \cdots h_t(u_1\# \cdots \#u_k) = \sigma(X_1)\# \cdots \#\sigma(X_k)$. □

### 2.7 Construction of the NFA $\mathcal{A}$ in quasilinear space

The input to Theorem 1 is given by three items: $A_\pm$, $\Omega$, and the pair of words $U, V \in (A_\pm \cup \Omega)^*$. The input size in bits is in

$$\mathcal{O}((|UV| + |A_\pm \cup \Omega|)\log(|A_\pm \cup \Omega|)).$$

Our non-deterministic procedure will use space $\mathcal{O}((|UV| + |A_\pm \cup \Omega|)\log(|UV| + |A_\pm \cup \Omega|))$, which is quasi-linear in the input size.

Let $n_0 = (|UV| + |A_\pm \cup \Omega|)\log(|A_\pm \cup \Omega|))$ be the input size. The definition of $n = n_{\mathrm{init}} = |W_{\mathrm{init}}|$ gives $n \in \mathcal{O}(n_0)$, and therefore we can also choose $n = n_{\mathrm{init}}$ as the new input size. In the preprocessing we transformed the original input into the word $W_{\mathrm{init}}$ over a larger set of variables. However, transforming the initial data into $W_{\mathrm{init}}$ is easily implementable in quasi-linear space.

The next observation is that, according to Definition 2, every well-formed word $W$ can be stored with $\mathcal{O}(n \log n)$ bits. Moreover, given $W \in M(B, \mathcal{X}, \theta, \mu)$, it can be checked (deterministically) in space $\mathcal{O}(n \log n)$ whether it is well-formed. The same is true for extended equations according to Definition 3.

Next, we can consider in space $\mathcal{O}(n \log n)$, one after another, all candidates $(W, B, \mathcal{X}, \theta, \mu) \xrightarrow{h} (W', B', \mathcal{X}, \theta', \mu')$ for arcs in $\mathcal{G}$. (We use that, by construction, $h$ can be encoded by a list of length $\mathcal{O}(n)$ for all arcs in $\mathcal{G}$.) Each time we check (deterministically) in space $\mathcal{O}(n \log n)$ whether there is indeed an arc $h$. If the answer is positive, we output the arc. The switch from the graph $\mathcal{G}$ to the NFA $\mathcal{A}$ is computationally easy: for every final vertex $(W', B, \emptyset, \emptyset, \mu)$ compute a prefix $\#w'\#$ of $W$ with $|w'|_{\#} = k$ and output the arc $(W', B, \emptyset, \emptyset, \mu) \xrightarrow{g_{w'}} \#$.

We used nondeterminism during the procedure, but this was not really necessary, as we could have produced an NFA satisfying Theorem 1 in deterministic quasi-linear space. However, such an NFA would contain a lot of unnecessary states and arcs.

In order to justify the last sentence in the abstract that deciding the existential theory of non-abelian free groups is in $\mathsf{NSPACE}(n \log n)$, (i.e., nondeterministic quasi-linear space) we need to check whether $L(\mathcal{A}) \neq \emptyset$. Thus we integrate this check into the construction of the NFA $\mathcal{A}$ right-away. So we modify our construction: instead of computing all vertices and all arcs of $\mathcal{G}$ we output vertices and arcs only if they belong to a path from an initial to a final vertex. More precisely, for each vertex $V$ we define a Boolean variable Useful$(V)$ which is 1 if $V$ is on a path from an initial to a final vertex and 0 otherwise. In order to compute Useful$(V)$ we guess such a path in nondeterministic space $\mathcal{O}(n \log n)$. Here we use the standard fact that the nondeterministic complexity class $\mathsf{NSPACE}(n \log n)$ is closed under complementation by Immerman–Szelepcsényi (1987), see [13]. Thus, our procedure outputs a vertex $V$ (resp. an arc $V = (W, B, \mathcal{X}, \theta, \mu) \xrightarrow{h} (W', B', \mathcal{X}, \theta', \mu') = V'$) only if Useful$(V) = 1$ (resp. Useful$(V) =$ Useful$(V') = 1$). If no vertex satisfies Useful$(V) = 1$, then we can output a one-state NFA without final states because then $W_{\text{init}}$ has no solution. If however, at least one vertex $V$ satisfies Useful$(V) = 1$, then the output is an NFA $\mathcal{A}$ which accepts a nonempty set of endomorphisms over $C$ and, according to our construction, $W_{\text{init}}$ has at least one solution; moreover, the reader is invited to show that $W_{\text{init}}$ has infinitely many solutions if and only if $L(\mathcal{A})$ is infinite.

### 2.8 Forward property of arcs

The previous section has shown how to produce the graph $\mathcal{G}$, which is now at our disposal. For every initial vertex $V_0$ with a given solution $(\mathrm{id}_{A^*}, \sigma_0)$, we need to establish the existence of a path $V_0 \xrightarrow{h_1} V_1 \cdots \xrightarrow{h_t} V_t$ to some final vertex $V_t = (W', B, \emptyset, \emptyset, \mu)$ such that the following equation holds

$$\sigma_0(W) = h_1 \cdots h_t(W'). \tag{8}$$

This relies on the following technical concept.

**Definition 4.** *Let $V = (W, B, \mathcal{X}, \theta, \mu) \xrightarrow{h} (W', B', \mathcal{X}', \theta', \mu') = V'$ be an arc in $\mathcal{G}$ and $(\alpha, \sigma)$ be a solution at $V$. We say that the tuple $(V \xrightarrow{h} V', \alpha, \sigma)$ satisfies*

*the* forward property *if there exists a solution* $(\alpha h, \sigma')$ *at* $V'$ *such that*

$$\alpha\sigma(W) = \alpha h\sigma'(W').$$

**Lemma 4.** *Let* $V = (W, B, \mathcal{X}, \theta, \mu) \xrightarrow{\varepsilon} (\tau(W), B, \mathcal{X}', \theta', \mu') = V'$ *be a substitution arc as in* ***4*** *or* ***6*** *and* $(\alpha, \sigma)$ *be a solution at* $V$*. Suppose that* $\sigma(X) = uv$ *and* $\tau(X) = uX$*. Then* $(V \xrightarrow{h} V', \alpha, \sigma)$ *satisfies the forward property.*

*Proof.* If we let $\sigma'(X) = v$ and $\mu'(X) = \mu(v)$ then we can write $\sigma = \tau\sigma'$ and the morphism $\sigma : M(B, \mathcal{X}, \theta, \mu) \to M(B, \theta, \mu)$ factorizes through $\sigma' : M(B, \mathcal{X}', \theta', \mu') \to M(B, \theta, \mu)$. □

**Lemma 5.** *Let* $V = (W, B, \mathcal{X}, \theta, \mu) \xrightarrow{\varepsilon} (\tau(W), B, \mathcal{X}, \theta', \mu) = V'$ *be a variable-typing arc as in* ***5*** *and* $(\alpha, \sigma)$ *be a solution at* $V$ *such that* $\sigma(X) \in c^*$*. (Thus, we have* $\mu(Xc) = \mu(cX) \in \mu(c^*)$*; and the arc with the new type* $(X, c) \in \theta'$ *is defined.) Then* $(V \xrightarrow{h} V', \alpha, \sigma)$ *satisfies the forward property.*

*Proof.* The morphism $\sigma : M(B, \mathcal{X}, \theta, \mu) \to M(B, \theta, \mu)$ factorizes canonically through $\sigma' : M(B, \mathcal{X}, \theta', \mu) \to M(B, \theta, \mu)$. □

**Lemma 6.** *Let* $V = (h(W), B, \mathcal{X}, \theta, \mu) \xrightarrow{h} (W', B', \mathcal{X}, \theta', \mu') = V'$ *be any compression arc as in* ***1****,* ***2*** *or* ***3*** *and* $(\alpha, \sigma)$ *be a solution at* $V$*. Suppose there exists a* $B'$*-solution* $\sigma'$ *at* $V'$ *such that* $\sigma : \mathcal{X} \to M(B, \theta, \mu)$ *factorizes through morphisms as follows*

$$\sigma : \mathcal{X} \xrightarrow{\sigma'} M(B', \theta', \mu') \xrightarrow{h} M(B, \theta, \mu).$$

*Then* $(\alpha h, \sigma')$ *is a solution at* $V'$ *with* $\alpha\sigma(W) = \alpha h\sigma'(W')$*. In particular,* $(V \xrightarrow{h} V', \alpha, \sigma)$ *satisfies the forward property.*

*Proof.* Trivial. □

It is clear that Lemma 6 does not suffice for our purpose. We cannot prevent $\sigma$ from using letters from $B$ which are not present in $B'$, but then no factorization $\sigma : \mathcal{X} \xrightarrow{\sigma'} M(B', \theta', \mu') \xrightarrow{h} M(B, \theta, \mu)$ exists if $h$ is induced by the identity, as in the case of alphabet reduction. As already mentioned in the main body of the text, we need this type of alphabet reduction only over empty type relations. We content ourselves with the following statement.

**Lemma 7.** *Let* $V = (W, B, \mathcal{X}, \emptyset, \mu) \xrightarrow{\varepsilon} (W', B', \mathcal{X}, \emptyset, \mu') = V'$ *be an alphabet reduction as in* ***3****, where* $B' \subsetneq B$ *and* $\mu'$ *is the restriction of* $\mu$*. Let* $(\alpha, \sigma)$ *be a solution at* $V$*. Define a* $B'$*-morphism* $\beta : M(B, \emptyset, \mu) \to M(B', \emptyset, \mu')$ *by* $\beta(b) = \alpha(b)$ *for* $b \in B \setminus B'$ *and define* $\sigma'(X) = \beta\sigma(X)$*. Then* $(\alpha h, \sigma')$ *is a solution at* $V'$ *with* $\alpha\sigma(W) = \alpha\varepsilon\sigma'(W')$*. In particular,* $(V \xrightarrow{h} V', \alpha, \sigma)$ *satisfies the forward property.*

*Proof.* Since $\alpha : M(B, \emptyset, \mu) \to M(A, \emptyset, \mu_0)$ is a morphism, we have $\mu\beta(b) = \mu\alpha(b) = \mu_0\alpha(b) = \mu(b)$ for all $b \in B \setminus B'$ and $\beta$ is indeed a morphism from $M(B, \emptyset, \mu)$ to $M(B', \emptyset, \mu')$.

Note that $M(B', \mathcal{X}, \emptyset, \mu')$ is a submonoid of $M(B, \mathcal{X}, \emptyset, \mu)$ and $\varepsilon$ realizes the inclusion of these free monoids. Hence $W = \varepsilon(W') = W'$ as words. In particular, $\sigma(W) = \sigma(\overline{W})$ implies $\sigma'(W') = \sigma'(\overline{W}')$. Thus, $(\alpha\varepsilon, \sigma')$ solves $V'$.

Finally, by definition of $\beta$ we have $\alpha = \alpha\beta$ because $\alpha$ is an $A$-morphism. Hence $\alpha = \alpha\varepsilon\beta$ and we obtain

$$\alpha\varepsilon\sigma'(W') = \alpha\varepsilon\sigma'(W) = \alpha\varepsilon\beta\sigma(W) = \alpha\sigma(W).$$

□

### 2.9 Compression.

This section finishes the proof of Theorem 1. Consider an initial vertex $V_0 = (W_{\text{init}}, A, \Omega, \emptyset, \mu_{\text{init}})$ with a solution $(\alpha, \sigma)$. We will show below that $\mathcal{G}$ contains a path $V_0 \xrightarrow{h_1} V_1 \cdots \xrightarrow{h_t} V_t$ to some final vertex $V_t = (W', B, \emptyset, \emptyset, \mu)$ such that $\sigma(W_{\text{init}}) = h_1 \cdots h_t(W')$, and so $\mathcal{G}$ contains all solutions to $W_{\text{init}}$. Let us show why then, indeed, we are almost done with Theorem 1. We augment the graph $\mathcal{G}$ by one more vertex which is just the symbol #. Recall that $\{X_1, \ldots, X_k\}$ has been the set of specified variables. Every final vertex $(W', B, \emptyset, \emptyset, \mu)$ has a unique factorization $W' = \#w'\#w''$ with $|w'|_{\#} = k$. Let us add arcs $(W', B, \emptyset, \emptyset, \mu) \xrightarrow{g_{w'}} \#$ where $g_{w'} : C^* \to C^*$ is the homomorphism (not necessarily respecting the involution) defined by $g_{w'}(\#) = w'$. If we define the NFA $\mathcal{A}$ as $\mathcal{G}$ with this augmentation and if we let # be the exclusive final vertex, then by Proposition 1 we obtain Theorem 1. The construction of $\mathcal{A}$ can easily be implemented by a nondeterministic procedure that uses $\mathcal{O}(n \log n)$ space. In order to show the existence of the path from $V_0$ to $V_t$ we apply the recompression method[1] of [5], but with a new and improved treatment of "block compression". We avoid solving linear Diophantine equations, and give a structural theorem involving EDT0L languages that is more precise than the result in [5].

We show the existence of the path corresponding to the solution $(\alpha, \sigma)$ using an alternation between "block compression" and "pair compression", repeated until we reach a final vertex. The procedures use knowledge of the solution being aimed for. We proceed along arcs in $\mathcal{G}$ of the form $V = (W, B, \mathcal{X}, \theta, \mu) \xrightarrow{h} V' = (W', B', \mathcal{X}', \theta', \mu')$ thereby transforming a solution $(\alpha, \sigma)$ to $V$ into a solution $(\alpha', \sigma')$ to $V'$. However, this is not allowed to be arbitrary: we must keep the invariant $\alpha\sigma(W) = \alpha' h \sigma'(W')$. For example, consider the alphabet reduction where $B' \subsetneq B$ and $W = W' \in (B' \cup \mathcal{X})^*$. In this case we have $h = \mathrm{id}_{C^*}$, which induces the inclusion $\varepsilon : M(B', \emptyset, \mu') \to M(B, \theta, \mu)$. If $\sigma$ does not use letters outside $B'$ there is no obstacle. In the other case, fortunately, we will need alphabet reduction only when the type relation is empty on both sides. Then we can define $\beta(b) = \alpha(b) \in A^*$ for $b \in B \setminus B'$ and $\beta(b) = b$ for $b \in B'$.

[1] Compression became a main tool for solving word equations thanks to [16].

We let $\sigma'(X) = \beta\sigma(X)$. This defines a $B'$-solution at $V'$. In some sense this is a huge "decompression" making $\sigma'(W)$ perhaps much longer than $\sigma(W)$. However, $(\alpha\varepsilon, \sigma')$ is a solution to $V'$.

A word in $w \in \Sigma^*$ is a sequence of *positions*, say $1, 2, \ldots, |w|$, and each position is labeled by a letter from $\Sigma$. If $W = u_0 x_1 u_1 \cdots x_m u_m$, with $u_i \in C^*$ and $x_i \in \Omega$, then $\sigma(W) = u_0\sigma(x_1)u_1 \cdots \sigma(x_m)u_m$ and the positions in $\sigma(W)$ corresponding to the $u_i$'s are henceforth called *visible*.

**Block compression** Let $V = (W, B, \mathcal{X}, \emptyset, \mu)$ be some current non-final vertex with an empty type relation and a solution $(\alpha, \sigma)$. We start a block compression only if $B \le |W| \le 29n$. Since $|C| = 100n$, there will be sufficiently many "fresh" letters in $C \setminus B$ at our disposal.

1. Follow arcs of type **4** and **6** to remove all variables with $|\sigma(X)| \le 2$. Thus, without restriction, we have $|\sigma(X)| > 2$ for all $X$. If $V$ became final, we are done and we stop. Otherwise, for each $X$ we have $\sigma(X) = bw$ for some $b \in B$ and $w \in B^+$. Following a substitution arc of type **6**, we replace $X$ by $bX$. (Of course, we also replace $\overline{X}$ by $\overline{Xb}$, changing $\mu(X)$ to $\mu(X) = \overline{\mu(\overline{X})} = \mu(w)$. From now on we always do this without further comment.) Every substitution $X \mapsto bX$ decreases $\sum_{X\in\mathcal{X}} |\alpha\sigma(X)|$, a fact which will be used later. Moreover, if $bX \le W$ and $b'X \le W$ are factors with $b, b' \in B$, then $\# \ne b = b'$ due to the previous substitution $X \mapsto bX$. For each $b \in B \setminus \{\#\}$ define sets $\Lambda_b \subseteq \mathbb{N}$ which contain those $\lambda \ge 2$ such that there is an occurrence of a factor $db^\lambda e$ in $\sigma(W)$ with $d \ne b \ne e$, where at least one of the $b$'s is visible. We also let $\mathcal{X}_b = \{X \in \mathcal{X} \mid bX \le W \wedge \sigma(X) \in bB^*\}$. Note that $\sum_b |\Lambda_b| + |\mathcal{X}_b| \le |W|$. Since $W$ is well-formed we have $\Lambda_b = \Lambda_{\overline{b}}$.
2. Fix some subset $B_+ \subseteq B$ such that for each $\# \ne b \in B$ we have $b \in B_+ \iff \overline{b} \ne B_+$. For each $b \in B_+$, where $\Lambda_b \ne \emptyset$, run the following *$b$-compression*:
3. $b$-compression. (This step removes all proper factors $b^\ell$ and $\overline{b}^\ell$, $\ell \ge 2$, from $W$.)
   (a) Introduce fresh letters $c_b, \overline{c_b}$ with $\mu(c_b) = \mu(b)$. In addition, for each $\lambda \in \Lambda_b$ introduce fresh letters $c_{\lambda,b}, \overline{c_{\lambda,b}}$ with $\mu(c_{\lambda,b}) = \mu(b)$. We abbreviate $c = c_b$, $\overline{c} = \overline{c_b}$, $c_\lambda = c_{\lambda,b}$, and $\overline{c_\lambda} = \overline{c_{\lambda,b}}$. We let $h(c_\lambda) = h(c) = b$ and we introduce a type by letting $\theta = \{(c_\lambda, c) \mid \lambda \in \Lambda_b\}$. Renaming arcs **1** realize this transformation. We did not touch $W = h(W)$, but we introduced partial commutation.
   (b) When we introduced $c, c_\lambda$ we did not change $W$, but we changed the alphabet $B$ to some larger alphabet $B'$. Now we change $W$ and its solution. We start to replace in $\sigma(W) \in B^*$ every factor $db^\lambda e$ (resp. $d\overline{b}^\lambda e$), where $d \ne b \ne e$ and $\lambda \in \Lambda_b$, with $dc^\lambda e$ (resp. $d\overline{c}^\lambda e$). This yields a new word $W' \in B'^*$, which was obtained via the renaming arc $h(c) = b$. Recall that for every $X \in \mathcal{X}_b$ we had $bX \le W$ and for some positive $\ell$ we had $\sigma(X) = b^\ell w$ with $w \notin bB^*$. In the new word $W'$ we have $cX \le W'$ and for the new solution $\sigma'$ we have $\sigma(X) = c^\ell w'$ with $w' \notin cB'^*$. We rename $W', B', \alpha' = \alpha h, \sigma'$ as $W, B, \alpha, \sigma$.

(c) We define $\theta = \{(c_\lambda, c) \mid \lambda \in \Lambda_b\} \cup \{(X, c) \mid X \in \mathcal{X}_b \wedge \sigma(X) \in c^*\}$. This can be realized by arcs **5**.

(d) Let $W \in M(B, \mathcal{X}, \theta, \mu)$ be given by some word in $W \in (B \cup \mathcal{X})^*$: scan the word $\sigma(W) \in B^*$ from left to right. Stop at each factor $dc^\lambda e$ with $d \neq c \neq e$ and $\lambda \in \Lambda_b$. If in this factor some position of the $c$'s is visible then choose exactly one of these visible positions and replace that $c$ by $c_\lambda$. If no $c$ is visible, they are all inside some $\sigma(X)$; then choose any $c$ and replace it by $c_\lambda$. Recall that $c$ and $c_\lambda$ commute, hence $dc^\lambda e$ became $dc_\lambda c^{\lambda-1}e = dc^{\ell_1}c_\lambda c^{\ell_2}e \in M(B, \theta, \mu)$ for all $\ell_1 + \ell_2 = \lambda - 1$. After that we run through the same steps for $\overline{c}$. The whole transformation can be realized by renaming arcs **1** defined by $h(c_\lambda) = c$. There is a crucial observation: if $X \in \mathcal{X}_b$ and we had $\sigma(X) = c^\ell w$ with $w \notin cB^*$ before the transformation then now still $\sigma'(X) = c^\ell w'$, but due to commutation $c_\lambda \sigma'(X)$ is a factor in $\sigma'(W') \in M(B, \theta, \mu)$. For example, assume $\overline{X}\overline{c}^2 Y c Z c X \overline{c}\overline{Y} d \leq W$ with $\sigma(X) = cd\overline{c}$, $\sigma(Y) = \overline{c}dc$, and $\sigma(Z) = c^2$. Then the corresponding factor in $W'$ looks as $\overline{X}\overline{c_4}\,\overline{c}Y c_6 Z c X \overline{c_4}\overline{Y} d = \overline{X}\overline{c_4}\,\overline{c}Y c_6 c Z X \overline{c_4}\overline{Y} d \in M(B, \mathcal{X}, \theta, \mu)$, but $c_6$ and $Z$ do not commute. However, it is important only that $\sigma'(Z) = c^2$ and $c_6$ commute, which they do.

(e) Rename $W', B', \alpha' = \alpha h, \sigma'$ as $W, B, \alpha, \sigma$. Perform the following loop 3(e)i – 3(e)iv until no $c$ and no $X \in \mathcal{X}_b$ with $\sigma(X) \in c^*$ occurs in $W$.

i. If $X \in \mathcal{X}_b$ and the maximal $c$-prefix of $\sigma(X)$ is odd then follow an substitution arc $X \mapsto cX$. Do the same for $\overline{b}$ and $\overline{c}$.

ii. If in the new solution $\sigma'(W')$ there is a factor $dc_\lambda c^\ell e$ with $\ell$ odd, then inside this factor the word $c_\lambda c$ is visible due to commutation and the previous step. In this case follow an arc **2** defined by $h(c_\lambda) = cc_\lambda$. Thus, w.l.o.g $\ell$ is even for all $dc_\lambda c^\ell e \leq \sigma'(W')$.

iii. Follow a compression arc defined by $h(c) = c^2$.

iv. Remove all $X$ with $\sigma'(X) = 1$ by following an substitution arc; and rename $W', B', \mathcal{X}', \alpha' = \alpha h, \sigma'$ as $W, B, \mathcal{X}, \alpha, \sigma$

(f) Let $B' = B \setminus \{c, \overline{c}\}$ and $\mu'$ be induced by $\mu$. Observe that no $c$ or $\overline{c}$ appears in $\sigma(W)$: they are all compressed into single letters $c_\lambda$. Thus the type relation of $B \cup \mathcal{X}'$ is empty again. Hence we can follow an alphabet reduction arc $(W, B, \mathcal{X}, \theta, \mu) \xrightarrow{\varepsilon} (W, B', \mathcal{X}', \emptyset, \mu')$. The new solution to $(W, B', \mathcal{X}', \emptyset, \mu')$ is the pair $(\alpha', \sigma)$ where $\alpha' = \alpha\varepsilon$ is defined by the restriction of $\alpha$ to $M(B', \emptyset, \mu')$.

Having performed $b$-compressions for all $b \in B_+$, we have increased the length of $W$. But it is not difficult to see that the total increase can be bounded in $\mathcal{O}(n)$. Actually, we have $|W| \leq 31n$ at the end because we started with $|W| \leq 29n$ and step 1 of block compression increases $|W|$ by at most $2n$, and no other step increases $|W|$. Now we use alphabet reduction in a final step of block compression in order to reduce the alphabet $B$ such that $|B| \leq |W|$. We end up at a vertex named again $V = (W, B, \mathcal{X}, \emptyset, \mu)$, which has a solution $(\alpha, \sigma)$. The difference to the situation before block compression is that now $|B| \leq |W| \leq 31n$, and no proper factor $b^2$, $b \in B$, can be found in $W$ anymore.

**Space requirements for the block compression** We start the block compression at a vertex $V = (W, B, \mathcal{X}, \emptyset, \mu)$ with a given solution $(\alpha, \sigma)$ only if $|B| \leq |W| \leq 29n$. For example, every initial vertex having a solution $(\mathrm{id}_{A^*}, \sigma)$ falls into that category. Now we recall all steps.

First we removed variables $X$ with $|\sigma(X)| \leq 2$, and for the remaining variables we did some substitution $X \mapsto bX$, which together increases the length by at most $2n$. Hence we reached a vertex $V' = (W', B, \mathcal{X}', \emptyset, \mu')$ with $|W'| \leq 31n$ via arcs satisfying the forward property by Lemma 4. Inspecting the procedure step by step, we verify whether each time we follow an arc if it satisfies the forward property using Lemmas 4, 5 and 6 (but without using Lemma 7 during the block compression). So wherever we stop, Equation (8) is valid during this compression procedure if it was valid before.

Next we have to show termination and also that we stay inside the graph $\mathcal{G}$ during the procedures. Termination can be seen as follows. Renaming arcs are used at most $\mathcal{O}(n)$ times, so they are irrelevant. No arc increases the sum $\sum_{x \in \mathcal{X}} |\sigma(X)|$. Either it decreases this sum, or if this sums remains stable, then $|W|$ decreases. This shows termination.

In order to show that we remain inside $\mathcal{G}$ we have to control the possible fluctuations of $|W|$ and $|B|$.

After step 1 we have $|B| \leq |W| \leq 31n$. Step 3(a) increases $|B|$ by introducing fresh letters $c_b, \overline{c_b}$ plus $\Lambda_b$. Since each such new letter corresponds to a visible letter in $W$, after this step we have $|B| \leq 62n$. No other steps increase $|B|$, and at the end we have $|B| \leq absW$.

Steps 2, 3(a)-(d) and 3(f) do not change the length of $W$. The only possible increase to $|W|$ can occur in step 3(e). Part 3(e)i may cause an increase of at most $2n$. Part 3(e)iii does not increase length, and part 3(e)iii decreases the length of $c, \overline{c}$-blocks by half. We now show that at step 3(f) we have $|W| \leq 31n$ again.

Let $\Theta_t = \{c_1^\dagger, \ldots, c_r^\dagger\}$ be the set of letters $c$ or $\overline{c}$ that are added to $W$ in the $t$-th iteration of step 3(e)i, so $r \leq 2n$. After step 3(e)i either

1. there is a distinct letter $c_i^\ddagger \in \{c, \overline{c}\}$ for each $c_i^\dagger \in \Theta_t$ so that $c_i^\dagger c_i^\ddagger \leq W$, in which case the compression $h(c) = cc$ in step 3(e)iii will replace $c_i^\dagger c_i^\ddagger$ by $c_i^\ddagger$,
2. there are factors in $W$ of the form $dc_\lambda \alpha c_{i_1}^\dagger \ldots c_{i_s}^\dagger e$ where $d, e \notin \{c, \overline{c}\}$ and $\alpha \in \{c^\varepsilon, \overline{c}^\varepsilon \mid 0 \leq \varepsilon < s\}$ (so each $c_{i_j}^\dagger$ does not have a distinct $c, \overline{c}$ letter matching it). In this case each $c_{i_j}^\dagger$ came from replacing a variable $X_j$ with $\sigma(X_j) = c_{i_j}^\dagger$. Some of the $c_{i_j}^\dagger$ letters will be consumed by steps 3(e)ii and iii, but if $s - \varepsilon \geq 2$ then some will remain. However for each such letter we have one less variable in $W$, so the number of such letters that can possibly be carried to the next iteration of step 3(e) during the entire procedure is bounded by $2n$.

So during each iteration of the loop, the number of letters increases in step 3(e)i by at most $2n$, and the number of $c^\dagger$ letters that remain after an iteration is at most $2n$, so $|W| \leq 31n + 4n = 35n$ throughout step 3(e). At the end of step 3(e)

all $c, \overline{c}$ letters have been consumed, which includes the letters $c^\dagger$ added in step 3(e)i, so $|W| \leq 31n$.

Hence we end the procedure with an increase in size which comes only from the first step: we end at some vertex $V' = (W', B', \mathcal{X}, \emptyset, \mu)$ where the type relation is empty again. Although $B'$ might be larger than $B$, we have $|B'| \leq |W'| \leq 31n$.

**Pair compression** After one round of block compression we run Jeż's procedure *pair compression*. It brings us back to $|B| \leq |W| \leq 29n$ and allows us to start another block compression. This is essential because it keeps the length in $\mathcal{O}(n)$.

Let us explain the procedure in detail. This is similar to [11], but some of the technical details are different. We roughly follow [5].

We start a pair compression at a vertex $V_p = (W, B, \mathcal{X}, \emptyset, \mu)$ with $|B| \leq |W| \leq 31n$, where $W$ contains no proper factor $b^2$ for any $b \in B$. Thus we can assume to have just performed a block compression. Moreover, we assume that $(\alpha, \sigma)$ is a solution to $V$. The goal is to end at a vertex $V_q = (W'', B'', \mathcal{X}', \emptyset, \mu''')$ with $|B''| \leq |W''| \leq 29n$ by some path satisfying the forward condition.

We begin with a random partition $B \setminus \{\#\} = L \cup R$ such that $b \in L \iff \overline{b} \in R$ which is constructed as follows. We first write $B \setminus \{\#\} = B_+ \cup \{\overline{b} \mid b \in B_+\}$ as a disjoint union. This is possible because $B \setminus \{\#\}$ has no self-involuting letters. Next, for each $b \in B_+$ we choose uniformly and independently whether either $b \in L$ and $\overline{b} \in R$, or $b \in R$ and $\overline{b} \in L$.

We have $ab \in LR \iff \overline{b}\overline{a} \in LR$, hence the compression respects the involution, and there is no overlap in $\sigma(W)$ between any occurrences of $ab \in LR$ and $cd \in LR$, unless it is the same occurrence and $ab = cd$. The idea is to compress in one phase all factors $ab \in LR$ into a single fresh letter. But the obstacle is that in $\sigma(W)$ an occurrence of some $ab \in LR$ can be "crossing". This means, there is some $aX \leq W$ (resp. $Xb \leq W$) with $\sigma(X) \in bB^*$ (resp. $\sigma(X) \in B^*a$). In this case, $\sigma(W)$ has an occurrence of a factor $ab$ where $a$ is visible, but $b$ is not visible. This forces us to "uncross" $ab$, which is a basic idea from [11] and appears in steps (2.) and (3.) below.

1. Remove all $X \in \mathcal{X}$ with $\sigma(X) = 1$ via substitution arcs.
2. Make $B'$ large enough such that $B \subseteq B' \subseteq C$ and for each $ab \in LR$ there exists a uniquely defined "fresh" letter $c_{ab} \in B' \setminus B$, subject to the condition that first, $ab \leq \sigma(W)$ and second, there is at least one occurrence of that factor such that either the position of $a$ or $b$ (or both) in $\sigma(W)$ is visible. Let us count how many fresh letters $c_{ab}$ we need. There are at most $31n$ positions in $\sigma(W)$ which are visible since $|W| \leq 31n$. Moreover, each visible position leads to at most one factor $ab \in LR$. Thus, the number of $c_{ab}$ is less than $31n$. Now, for each $c_{ab}$ we let $\overline{c_{ab}} = c_{\overline{b}\overline{a}}$. Let us argue that $c_{\overline{b}\overline{a}}$ is present. This is clear if $ab \leq W$ because in that case $\overline{b}\overline{a} \leq W$, too. Otherwise some $a$ is visible and $aX \leq W$ for a variable $X$. We obtain $\overline{X}\overline{a} \leq W$, so the position of $\overline{a}$ is visible in $\sigma(W)$, and, hence, $\overline{b}\overline{a} \leq \sigma(W)$ creates the letter $c_{\overline{b}\overline{a}}$. Thus, $|B' \setminus B| \leq 31n$ and therefore $|B'| \leq 62n$.

The following always holds: $ab \neq \overline{b}\overline{a}$. If we assume the contrary, then a factor $ab = a\overline{a}$ appears in $\sigma(W)$ and suppose $a$ is visible. We cannot have $a\overline{a} \leq W$ because in a well-formed word there are no proper factors $w$ with $\mu(w) = 0$. Hence, (by symmetry) we must have $aX \leq W$ for some $X$ with $\sigma(X) = \overline{a}w'$ and, as a consequence, $\mu(X) = \mu(\overline{a})\mu(w') = (\overline{a}, \overline{a}) \cdot \mu(w') \in N$. Thus, $\mu(aX) = 0$, but $aX$ is a proper factor of $W$, contradiction. Note the rather far reaching consequence of this last tiny computation: as $ab \neq \overline{b}\overline{a}$ we can compress later unambiguously $ab$ into $c_{ab}$ and $\overline{b}\overline{a}$ into $\overline{c_{ab}}$ without creating any self-involuting letter. Thus, $c_{ab} \neq \overline{c_{ab}}$ and we maintain the invariant that no other symbol than # is self-involuting.
We realize this alphabet enlargement, from $B$ to $B'$, via compression arcs labeled by $h(c_{ab}) = ab$. So far we have not changed $W$. We simply followed compression arcs which satisfy the forward property. We are now at some vertex $V = (W, B', \mathcal{X}', \emptyset, \mu')$ with the solution $(\alpha h, \sigma)$. Indeed, $\alpha h \sigma(W) = \alpha\sigma(W)$ as $\sigma(W) \in B^* \subseteq B'^*$.

3. Create a list $\mathcal{L} = \{X \in \mathcal{X}' \mid \exists b \in R : \sigma(X) \in bB^*\}$. For each $X \in \mathcal{L}$ do in any order:
   – if $\sigma(X) \in bB^*$ with $b \in R$ then follow a substitution arc $X \mapsto bX$.
   Remember, if we follow $X \mapsto bX$ then automatically $\overline{X}$ is replaced with $\overline{X}\,\overline{b}$, too; and $\overline{b} \in L$. We also have $\{X, \overline{X}\} \subseteq \mathcal{L}$ if and only if $\sigma(X) \in bB^*a$ for some $ab \in LR$. In that case we actually substituted $X$ by $bXa$ and $\overline{X}$ by $\overline{a}\overline{X}\,\overline{b}$. In any case, we have successfully "uncrossed" every $ab \in LR$. We followed substitution arcs which satisfy the forward property by Lemma 4. We are now at a vertex $V' = (W', B', \mathcal{X}', \emptyset, \mu'')$ with a solution $(\alpha h, \sigma')$. If $ab \leq \sigma'(W')$ then in every occurrence of $ab$ either both positions $\sigma'(W')$ in are visible or neither are. This concludes the "uncrossing".
4. For each $ab \in LR$ such that $c_{ab}$ was generated follow a compression arc labeled by $h(c_{ab}) = ab$: we replace all occurrences of $ab$ in $\sigma'(W')$ by the letter $c_{ab}$, and simultaneously replace all occurrences of $\overline{b}\overline{a}$ in $\sigma'(W')$ by the letter $\overline{c_{ab}}$, with no overlapping ambiguity. It is therefore clear that the solution $\sigma'$ factorizes through $h$. Due to the previous "uncrossings", all arcs satisfy the forward property. Note that $h^2 = h$ because $h : B' \to B'$ is a $B$-morphism and $a, b \in B$. Hence, we are now at a vertex $V' = (W'', B', \mathcal{X}', \emptyset, \mu''')$ with $W' = h(W'')$ which has a solution $(\alpha h, \sigma'')$ where $\sigma'' = h\sigma'$.
5. Finally, we perform an alphabet reduction as in **3** which replaces $B'$ by some smaller alphabet $B''$. As $\#a\# \leq W''$ (we never touched any occurrence of $\#a\#$) we see that automatically $A \subseteq B''$. The alphabet reduction is done when the type relation is empty. Thus, we use Lemma 7 to see that we can realize this step by an arc which satisfies the forward property. We achieved our goal of reaching a vertex $V_q = (W'', B'', \mathcal{X}', \emptyset, \mu''')$ with a solution of the form $(\alpha h, \beta\sigma'')$. This finishes the procedure *pair compression*.

The termination of the procedure is immediate. The maximal number of steps is bounded by $\mathcal{O}(n)$. Suppose $V_p = (W, B, \mathcal{X}, \emptyset, \mu)$ is the start vertex and $V_q = V'' = (W'', B', \mathcal{X}', \emptyset, \mu''')$ is the endpoint. Denote this path by

$$V_p \xrightarrow{h_{p+1}} V_{p+1} \xrightarrow{h_{p+2}} V_{p+2} \xrightarrow{h_{p+3}} \cdots \xrightarrow{h_q} V_q.$$

Along the path we used only arcs satisfying the forward condition. Thus, starting with a solution $(\alpha, \sigma)$ we find some solution $(\alpha h_{p+1} \cdots h_q, \sigma'')$ to $V_q$ such that

$$\alpha\sigma(W) = \alpha h_{p+1} \cdots h_q \sigma''(W'').$$

Thus we maintained the validity of Equation (8) throughout all iterations of block and pair compressions.

What remains to be shown is that we can achieve $|W''| \leq 29n$ for at least one partition $B \setminus \{\#\} = L \cup R$.

We reformulate the probabilistic argument of [11] in our setting. We started a pair compression at a vertex $V_p = (W, B, \mathcal{X}, \emptyset, \mu)$ with $|W| \leq 31n$. Let us factorize the word $W \in (B \cup \mathcal{X})^+$ as $W = x_0 u_1 x_1 \cdots u_m x_m$ such that

1. $u_i \in (B \setminus \{\#\})^+$ for $1 \leq i \leq m$, i.e., each $u_i$ is a nonempty word over constants,
2. The length of each $|u_i|$ is divisible by 3,
3. $x_i \in (B \cup \mathcal{X})^*$ for $0 \leq i \leq m$,
4. $|x_0 \cdots x_m| \leq 3n$.

This is possible because $|W|_\# + \sum_{X \in \mathcal{X}} |W|_X \leq n$. We need $|x_0 \cdots x_m| \leq 3n$ rather than $n$ because we must adjust the lengths of the $x_i$'s in order to guarantee divisibility by 3 of the $|u_i|$'s. By inserting factors of the form $x_i = 1$ we may assume:

$$|u_i| = 3 \quad \text{for all } 1 \leq i \leq m, \tag{9}$$

$$|W| = |x_0 \cdots x_m| + 3m. \tag{10}$$

Consider the word $W'$ which was obtained by the substitution arcs, but before the compression of factors $ab \in LR$ into single letters. The increase in length $|W'| - |W|$ is due to substitution arcs $X \mapsto bX, \overline{X} \mapsto \overline{X}\,\overline{b}$ with $X \in \mathcal{L}$, so the length goes up by $2n$. Note that the $u_i$ factors do not change, only the $x_i$ factors. Hence, $W'$ has the factorization $W' = y_0 u_1 y_1 \cdots u_m y_m$ with $y_i \in (B \cup \mathcal{X})^*$ and

$$|y_0 \cdots y_m| \leq |x_0 \cdots x_m| + 2n. \tag{11}$$

Let $W''$ be the word after pair compression has been performed. So $|W''| \leq |W'|$. If $|W| \leq 27n$ then

$$|W''| \leq |W'| = |y_0 \ldots y_m| + |u_1 \ldots u_m| \leq |x_0 \ldots x_m| + 2n + |u_1 \ldots u_m|$$

$$= |x_0 u_1 \ldots u_m x_m| + 2n = |W| + 2n \leq 27n + 2n = 29n$$

and we are done.

Hence, let us assume $27n \leq |W|$. We have $27n \leq |W| = |x_0 \ldots x_m| + 3m \leq 3n + 3m$ which means $m \geq 8n$.

The word $W''$ is the compression of a word $y_0 v_1 y_1 \cdots v_m y_m$ where each $v_i$ is the result of the compression restricted to $u_i$. Each $u_i$ can be written as $u_i = abc$ with $a, b, c \in B$. Since $W$ did not contain any proper factor $d^2$ with

$d \in B$, (as we have performed block compression first) we know $a \neq b \neq c$. There are two possibilities: either $b \in L$ or $b \in R$. In the first case, either $c \in R$ or $c \in L$, and in the second case either $a \in L$ or $a \in R$. Each event $bc \in LR, bc \in LL, ab \in LR, ab \in RR$ has probability $\frac{1}{4}$, so with probability $\frac{1}{2}$ $u_i$ is compressed from length 3 to 2, and with probability $\frac{1}{2}$ it remains length 3. Thus, for the expectation we obtain $E[|v_i|] = \frac{2}{2} + \frac{3}{2} = \frac{5}{2}$. By linearity of expectation, we obtain

$$E[|v_1 \cdots v_m|] = \tfrac{5}{2}m. \tag{12}$$

Since the expected length is $\frac{5}{2}m$, and $m$ is even, there must exist at least one partition $B \setminus \{\#\} = L \cup R$ which satisfies $|v_1 \cdots v_m| \leq \frac{5}{2}m$.

So, we "change our algorithm" and force the algorithm to choose exactly this partition (in other words, there is a path in the graph which chooses this partition, so we follow this one). We can thus guarantee $|v_1 \cdots v_m| \leq \frac{5}{2}m$ by this choice. We may estimate the length of $W''$ as follows.

$$\begin{aligned}
|W''| &\leq |y_0 \cdots y_m| + |v_1 \cdots v_m| \\
&\leq |x_0 \cdots x_m| + 2n + \tfrac{5}{2}m \\
&= |W| + 2n - \tfrac{m}{2} && \text{since } |W| = |x_0 \cdots x_m| + 3m \\
&\leq |W| - 2n && \text{since } m \geq 8n \\
&\leq 29n && \text{since } |W| \leq 31n.
\end{aligned}$$

Due to $|W''| \leq 29n$, we can run another block compression, then a pair compression and so on. We alternate between block and pair compressions inside the graph $\mathcal{G}$, always following arcs satisfying the forward condition.

**Compression terminates** Let $s(0) = |W_{\text{init}}|$ and $s(i)$ be the word obtained from $s(i-1)$ by a single of application of block then pair compression.

A priori, although we have $s(i) \leq 29n$ for all $i \geq 0$ the compression method could run forever. Let us show that this can never happen.

**Lemma 8 (Termination).** *The alternation between block and pair compression terminates with some final vertex.*

*Proof.* By contradiction assume the contrary. Then there exists an infinite path alternating between block and pair compressions satisfying the forward condition. However, due to the first step in the block compression this infinite path uses infinitely many substitution arcs $X \mapsto bX$, where $b$ is a constant. As all arcs satisfy the forward condition for each substitution, each use of an arc $X \mapsto bX$ decreases $\sum_{x \in \mathcal{X}} |\alpha\sigma(X)|$. No use of any arc increases this sum. Since we started with a fixed solution $(\mathrm{id}_{A*}, \sigma_0)$ at some initial vertex, there are no such infinite paths. This is a contradiction, and Theorem 1 is shown. □

## PART II. Proof of Theorem 2

This part contains some repetitions of what has been written above. We hope that these redundancies make the reading easier. More importantly, the set-up for Theorem 2 is more general, and this generates additional technical arguments. The proof of Theorem 2 is therefore more technical and more difficult than that of Theorem 1 because we allow more general constraints, and we have to cope with the elements of order 2 which appear in the free products – as for example in the modular group.

## 3 Preliminaries for the proof of Theorem 2

### 3.1 Free products: special features of $\mathbb{F}$

Our results hold for finitely generated free products

$$\mathbb{F} = \star_{1 \le i \le p} F_i$$

where each $F_i$ satisfies one of the following conditions:

- $F_i = \mathrm{F}(A_i)$ is a free group with basis $A_i$.
- $F_i = A_i^*$ is a free monoid with involution over a set $A_i$ with involution. (Recall that the involution on $A_i$ might be the identity. Hence, if $A_i^*$ is the free monoid, then the involution means reading words from right-to-left.)
- $F_i$ is a finite group. (Every group is viewed as a monoid with involution by defining $\overline{x} = x^{-1}$.)

The monoid $\mathbb{F}$ is a monoid with an involution that is induced by the involutions on each $F_i$. It is not essential that the finite monoids $F_i$ are groups, but it simplifies the presentation as there are fewer cases. By $U(\mathbb{F})$ we denote the submonoid of *units*, that is, the invertible elements in $\mathbb{F}$. Thus $U(\mathbb{F})$ is the free product over those $F_i$ which are groups. To make our results nonvacuous we assume that $\mathbb{F}$ is infinite. Also, we assume that the multiplication table for each finite group $F_i$ is part of the input.

Given $\mathbb{F}$, we choose as a set of monoid generators the smallest subset $A_{\mathbb{F}} \subseteq \mathbb{F}$ which is closed under involution and which contains the sets $A_i$ (if $F_i = \mathrm{F}(A_i)$ or $F_i = A_i^*$) and all sets $F_i \setminus \{1\}$ where $F_i$ is finite. We let $\pi : A_{\mathbb{F}}^* \to \mathbb{F}$ be the canonical morphism.

If $\mathbb{F}$ is a group, then $\mathbb{F}$ is a finitely generated free product of infinite cyclic and finite groups. These groups are also known as *plain groups* or as *basic groups*. Basic groups form a proper subclass of the class of *virtually free groups* which are those groups having a free subgroup of finite index.[2] A geodesic triangle in the Cayley graph of a plain group, with respect to the standard generators, is depicted in Figure 2.

[2] For example, the modular group $\mathrm{PSL}(2,\mathbb{Z}) = \mathbb{Z}/2\mathbb{Z} \star \mathbb{Z}/3\mathbb{Z}$ is plain. The group $\mathrm{SL}(2,\mathbb{Z})$ is isomorphic to the amalgamated product $\mathbb{Z}/4\mathbb{Z} \star_{\mathbb{Z}/2\mathbb{Z}} \mathbb{Z}/6\mathbb{Z}$. Hence, it is virtually free, but it is not plain because it is infinite and has a non-trivial center.

A word $w \in A_{\mathbb{F}}^*$ is called *reduced* if it is the shortest word representing the element $\pi(w) \in \mathbb{F}$. (These words are also called *geodesics* in the literature.) A word $w$ is reduced (resp. geodesic) if and only if for each factor $ab \leq w$ with $a, b \in A_{\mathbb{F}}$ we have first, if $a \in \mathrm{F}(A_i)$ then $b \neq a^{-1}$ and second, if $a \in F_i$ and if $F_i$ is finite then $b \notin F_i$.

We identify $\mathbb{F} \subseteq A_{\mathbb{F}}^*$ with the regular set of reduced words in $A_{\mathbb{F}}^*$. Note that every word in a free monoid $A_i^*$ is reduced, so factors of the form $a\overline{a}$ may appear in reduced words. Moreover, if $F_i$ is finite, then there might be elements $x \neq 1 = x^2$. Hence, the equation $X^2 = 1$ may have a non-trivial solution. These are the main reasons why the proof of Theorem 2 is more involved than the proof of Theorem 1.

The monoid for recognizing the regular subset of reduced words is almost the same finite monoid as defined in (2). We replace (2) by the following monoid $N_{\mathbb{F}} = \{1, 0\} \cup A_{\mathbb{F}} \times A_{\mathbb{F}}$, where $1 \cdot x = x \cdot 1 = x$, $0 \cdot x = x \cdot 0 = 0$, and

$$(a, b) \cdot (c, d) = \begin{cases} 0 & \text{if } bc \text{ is reduced} \\ (a, d) & \text{otherwise} \end{cases} \tag{13}$$

The morphism $\psi_{\mathbb{F}} : A_{\mathbb{F}}^* \to N_F$ defined by $\psi_{\mathbb{F}}(a) = (a, a)$ for $a \in A_{\mathbb{F}}$ recognizes $\mathbb{F} \subseteq A_{\mathbb{F}}^*$.

Note that the group of units is a rational subset of $\mathbb{F}$. We view $U(\mathbb{F}) \subseteq \mathbb{F} \subseteq A_{\mathbb{F}}^*$ and we let $U = \{a \in A_{\mathbb{F}} \mid a \in U(\mathbb{F})\}$. We use the monoid $\mathbb{B} = \{1, 0\}$ in order to recognize $U \cup \{1\}$. The recognizing morphism $\psi_U$ maps $\{1\} \cup U$ to 1 and all other letters to 0. Now, consider the monoid

$$N = (\{1, 0\} \cup A_{\mathbb{F}} \times A_{\mathbb{F}}) \times \mathbb{B}.$$

Define $\psi : A_{\mathbb{F}}^* \to N$ by $\psi(a) = (\psi_{\mathbb{F}}(a), \psi_U(a))$. Hence,

$$\psi(a) = \begin{cases} ((a, a), 0) & \text{if } a \in A_{\mathbb{F}} \setminus U \\ ((a, a), 1) & \text{if } a \in U \end{cases} \tag{14}$$

Then $\psi$ recognizes $U(\mathbb{F})$ and $\mathbb{F}$ simultaneously. This works because

$$U(\mathbb{F}) = \mathbb{F} \setminus A_{\mathbb{F}}^* \cdot (A_{\mathbb{F}} \setminus U) \cdot A_{\mathbb{F}}^*.$$

Moreover, if $A_{\mathbb{F}}$ is embedded in any larger alphabet $A$ then, by extending $\psi$ to a homomorphism $\psi : A* \to N$ by $\psi(a) = (0, 0)$ for all $a \in A \setminus A_{\mathbb{F}}$, we obtain that $\psi$ recognizes simultaneously all Boolean combinations of the sets $\{1\}$, $U(\mathbb{F})$, $\mathbb{F}$, and $A_{\mathbb{F}}^*$ inside $A^*$. The monoid $N$ has an efficient representation and $|N| \leq 2(2 + |A_{\mathbb{F}}|^2)$.

In her thesis Michèle Benois proved that the family $\mathrm{RAT}(\mathbb{F})$ forms an effective Boolean algebra. Her statement in [3] is for free groups only, but her proof holds for the free product $\mathbb{F}$, too.

**Proposition 2 (Benois, 1969).** *Let $R \in \mathrm{RAT}(\mathbb{F})$ be rational and $R = \pi(L(\mathcal{A}))$, where $\mathcal{A} = (Q, A_{\mathbb{F}}^*, \delta, I, F)$ is an NFA with $n$ states over the free monoid $A_{\mathbb{F}}^*$ and $\delta \subseteq Q \times (A_{\mathbb{F}} \cup \{1\}) \times Q$. Then there exists an NFA $\mathcal{A}'$ with $n$ states satisfying*

$$R = \pi(L(\mathcal{A}')) \subseteq L(\mathcal{A}').$$

*In particular,* $\mathrm{RAT}(\mathbb{F})$ *forms an effective Boolean algebra.*

*Proof.* (Sketch) For $p, q \in Q$ let $L(p,q)$ denote the set of words labeling a path in $\mathcal{A}$ from $p$ to $q$. We construct an automaton $\mathcal{A}' = (Q, A_{\mathbb{F}}^*, \delta', I, F)$ by defining $\delta'$ as follows. Set $\delta' = \delta$. Repeat the following loop: as long as there are $a, b \in A_{\mathbb{F}}$ such that $ab \neq \pi(ab) = c \in A_{\mathbb{F}} \cup \{1\}$ with $ab \in L(p,q)$ but $(p,c,q) \notin \delta'$, replace $\delta'$ by $\delta' \cup \{(p,c,q)\}$.

This process takes at most $|Q|^2 (|A_{\mathbb{F}}| + 1)$ steps before it terminates; and it produces an NFA $\mathcal{A}'$ as desired.

In order to show that $\mathrm{RAT}(\mathbb{F})$ forms an effective Boolean algebra, it is enough to show that it is effectively closed under complementation. Therefore let $\mathcal{A}''$ be an NFA accepting the complement $A_{\mathbb{F}}^* \setminus L(\mathcal{A}')$. Since $\mathbb{F}$ is a regular subset of $A_{\mathbb{F}}^*$, the set $L(\mathcal{A}'') \cap \mathbb{F}$ is regular, hence rational. We have $\pi(L(\mathcal{A}'') \cap \mathbb{F}) = \mathbb{F} \setminus R$. Thus, $\mathbb{F} \setminus R \in \mathrm{RAT}(\mathbb{F})$ since $\mathbb{F} \setminus R$ is the homomorphic image of a rational set. □

Proposition 2 is crucial: it is the (only) justification for our convention that rational constraints $X \in R$ for $\mathbb{F}$ are specified by $X \in \rho^{-1}(m)$ where $\rho : A_{\mathbb{F}}^* \to N$ is a homomorphism to a finite monoid $N$ and $m \in N$. Recall that for a given morphism $\sigma : \Omega \to A_{\mathbb{F}}^*$ the semantics is $\rho\sigma(X) = m$. It is essential to have a one-to-one correspondence between $\mathrm{RAT}(\mathbb{F})$ and the set $\{L \in \mathrm{RAT}(A_{\mathbb{F}}^*) \mid L \subseteq \mathbb{F}\}$, and this is induced by the homomorphism $\pi : A_{\mathbb{F}}^* \to \mathbb{F}$.

### 3.2 From $\mathbb{F}$ to the free monoid $A_{\mathbb{F}}^*$ with involution.

EDT0L languages are closed under finite unions. Making non-determinstic guesses (which cover all cases) and pushing negations to atomic formulas we may assume without restriction that the input $\Phi$ is given a conjunction of atomic formulas of either type: $U = V$, $U \neq V$, $X \in \rho^{-1}(m)$, and $X \notin \rho^{-1}(m)$. Recall that $\rho : A_{\mathbb{F}}^* \to N$ is a homomorphism to a finite monoid. Since we may assume that $N$ has at least two elements, we can replace each subformula $X \notin \rho^{-1}(m)$ by $X \in \rho^{-1}(m')$, where $m \neq m'$ since, by definition, $X \in \rho^{-1}(m)$ refers to an evaluation over $A_{\mathbb{F}}^*$ and not over $\mathbb{F}$.

Concerning $U = V$ and $U \neq V$, we use standard triangulation and obtain the following equations and inequalities: $X \neq Y$, $X = yz$, and $X = 1$ with $X, Y \in \Omega$ and $|x| = |y| = 1$. Without restriction we can assume that $\rho^{-1}(1) = \{1\}$, hence we can replace $X = 1$ by the constraint $X \in \rho^{-1}(1)$. Thus only equations $X = yz$ and inequalities $X \neq Y$ remain.

Next, we follow a well-known procedure for replacing these equations and inequalities over $\mathbb{F}$ by equivalent formulas over $A_{\mathbb{F}}^*$, using equations and rational constraints, only. For an equation $X = yz$ we use the following equivalence, which is true for all reduced words $x, y, z \in \mathbb{F} \subseteq A_{\mathbb{F}}^*$:

$$\begin{aligned} x = \pi(yz) \iff \exists P, Q, R\, \exists a, b, c \in A_{\mathbb{F}} \cup \{1\} :\ & R \in U(\mathbb{F}) \wedge \\ & a = \pi(bc) \wedge x = PaQ \wedge y = PbR \wedge z = \overline{R}cQ \wedge R \in U(\mathbb{F}). \end{aligned}$$

For a "visual" proof of this equation see Figure 2.

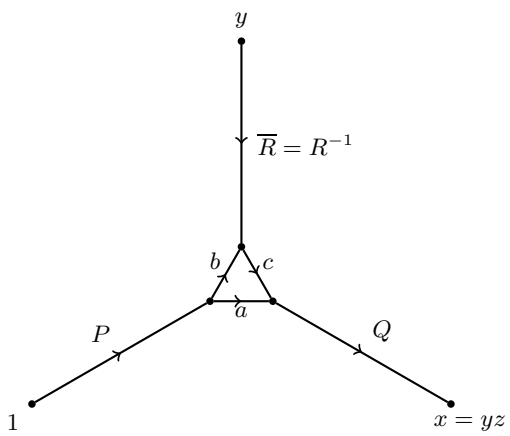



**Figure2.** Part of the Cayley graph of $\mathbb{F}$: the geodesic triangle to an equation $x = \pi(yz)$.

For an inequality $X \neq Y$ we use the following equivalence, which is true for all words $x, y \in A_{\mathbb{F}}^*$ (it is here that we use the assumption that $\mathbb{F}$ is infinite):

$$x, y \in \mathbb{F} \wedge x \neq y \iff$$
$$\exists P, Q, R \, \exists a, b, c \in A_{\mathbb{F}} : b \neq c \wedge xa = PbQ \wedge ya = PcR \wedge xa \in \mathbb{F} \wedge ya \in \mathbb{F}.$$

*Remark 1.* We have shown that it is enough to prove Theorem 2 in the special case where $\mathbb{F} = A_{\mathbb{F}}^*$ is a free monoid with involution. In particular, all words are reduced. The formula $\Phi$ is a conjunction of triangular equations $X = yz$ and of constraints $X \in \rho^{-1}(m)$. The main remaining obstacle is that $A_{\mathbb{F}}$ might contain self-involuting reduced words. In particular, there can be a letter $a$ with $a = \overline{a}$, and even if there is no such letter then still $a\overline{a}$ is a self-involuting reduced word.

### 3.3 How to remove self-involuting letters

The alphabet $A_{\mathbb{F}}$ is an alphabet with involution, denoted here by $\widetilde{\ }$. As we are in the general situation we might have $\widetilde{a} = a$ for some letter $a \in A_{\mathbb{F}}$. Choose some subset $A_+ \subseteq A_{\mathbb{F}}$ such that the size of $A_+$ is minimal while satisfying

$$A_{\mathbb{F}} = A_+ \cup \{\widetilde{a} \mid a \in A_+\}.$$

By minimality, we have

$$A_+ \cap \{\widetilde{a} \mid a \in A_+\} = \{a \in A_{\mathbb{F}} \mid a = \widetilde{a}\}.$$

We let $A_-$ be a disjoint copy of $A_+$. We can write $A_- = \{\overline{a} \mid a \in A_+\}$ and then $\overline{\overline{a}} = a$ makes $A_\pm = A_+ \cup A_-$ into an alphabet with involution $\overline{\ }$ without self-involuting letters. Now we encode words over $A_{\mathbb{F}}^*$ as words over $A_\pm^*$ via a

morphism $\iota : A_{\mathbb{F}}^* \to A_{\pm}^*$, where $\iota(a) = a$ and $\iota(\widetilde{a}) = \overline{a}$ if $\widetilde{a} \neq a \in A_+$, and $\iota(a) = a\overline{a}$ if $\widetilde{a} = a$. Note that $\iota$ respects the involution. (The morphism $\iota$ is a *code* because $\iota(A_{\mathbb{F}})$ is the basis of a free submonoid in $A_{\pm}^*$.) We also define a homomorphism $\eta : A_{\pm} \to A_{\mathbb{F}}$ in the other direction by defining $\eta(a)$ and $\eta(\overline{a})$ for $a \in A_+$ as follows. We let $\eta(a) = a$ and if $a \neq \widetilde{a}$ then $\eta(\overline{a}) = \widetilde{a}$. If, however, $a = \widetilde{a}$, then $\eta(\overline{a}) = 1$. The homomorphism $\eta$ does not respect the involution, but this does no harm. Note also that $\eta$ erases letters. However, the restriction of $\eta$ to a subset of $\iota(A_{\mathbb{F}}^*)$ injective.

We transform $\Phi$ in Remark 1 as follows. An equation $U = V$ is replaced by $\iota(U) = \iota(V)$, and every variable $X$ receives an additional constraint $X \in \iota(A_{\mathbb{F}}^*)$. Note that $\iota(A_{\mathbb{F}}^*)$ is a regular subset in $A_{\pm}^*$ and we can recognize $\iota(A_{\mathbb{F}}^*)$ by $\rho_{A_{\mathbb{F}}} : A_{\pm}^* \to \mathbb{N}_{A_{\mathbb{F}}}$, where $N_{A_{\mathbb{F}}}$ has size $\mathcal{O}(n^2)$. The precise definition of $\rho_{A_{\mathbb{F}}}$ is natural and left to the reader. Previous constraints $X \in \rho^{-1}(m)$ are replaced by $X \in \eta^{-1}(\rho^{-1}(m))$. Thus, we have transformed $\Phi$ over $A_{\mathbb{F}}^* \cup \Omega$ into a new formula $\Phi'$ over $A_{\pm} \cup \Omega$. Moreover, the construction guarantees that $\eta$ defines a bijection between $\mathrm{Sol}(\Phi')$ and $\mathrm{Sol}(\Phi)$. Thus, if $\mathrm{Sol}(\Phi') = \{\varphi(\#) \mid \varphi \in \mathrm{L}(\mathcal{A}')\}$, then we find another NFA $\mathcal{A}$ which has just one more state than $\mathcal{A}'$, and we obtain:

$$\mathrm{Sol}(\Phi) = \eta(\mathrm{Sol}(\Phi') = \{\eta\varphi(\#) \mid \varphi \in \mathrm{L}(\mathcal{A}')\} = \{\psi(\#) \mid \psi \in \mathrm{L}(\mathcal{A})\}\,.$$

*Remark 2.* By the equation above, Remark 1, and the last transformations, it is now enough to prove Theorem 2 in the special case where $\mathbb{F} = A_{\pm}^*$ is a free monoid with involution and $\Phi$ is a conjunction of equations and constraints of the form $X \in \rho^{-1}(m)$. Moreover, $A_{\pm}$ is a set where the involution has no fixed points.

### 3.4 How to force recognizing homomorphisms to respect the involution

Assume the homomorphism $\rho : A_{\mathbb{F}}^* \to N$ defining the rational constraint in Remark 1 was in fact a morphism to a finite monoid with involution. This property could have been lost during the transformation leading to Remark 2.

We now replace a recognizing homomorphism by a recognizing morphism to a finite monoid with involution. We show that there is a natural embedding of monoids (with or without involution) into monoids with involution. If $M$ is any monoid, then we define its dual monoid $M^T$ to be based on the same set $M^T = M$ as a set, where $M^T$ is equipped with a new multiplication $x \circ y = yx$. In order to indicate whether we view an element in the monoid $M$ or $M^T$, we use a flag: for $x \in M$ we write $x^T$ to indicate the same element in $M^T$. Thus, we can suppress the symbol $\circ$ and we simply write $x^T y^T = (yx)^T$. The notation is intended to mimmick transposition in matrix calculus. Similarly, we frequently write 1 instead of $1^T$ which is true for the identity matrix as well. Now the direct product $M \times M^T$ becomes a monoid with involution by letting $\overline{(x, y^T)} = (y, x^T)$. Indeed,

$$\overline{(x_1, y_1^T) \cdot (x_2, y_2^T)} = (y_2 y_1, (x_1 x_2)^T) = \overline{(x_2, y_2^T)} \cdot \overline{(x_1, y_1^T)}.$$

The following items are essential.

– If $M$ is finite then $M \times M^T$ is finite, too.
– We can embed $M$ into $M \times M^T$ by a homomorphism $\iota : M \to M \times M^T$ defined by $\iota(x) = (x, 1)$. Note that if $\eta : M \times M^T \to M$ denotes the projection onto the first component, then $\eta\iota = \mathrm{id}_M$. In particular, every homomorphism $\rho : M \to N$ of monoids factorizes through $\iota\rho : M \to N \times N^T$. We have $\rho = \eta\iota\rho$.
– If $M$ is a monoid with involution and $\rho : M \to N$ is a homomorphism of monoids, then we can lift $\rho$ uniquely to a morphism $\mu : M \to N \times N^T$ of monoids with involution such that we have $\rho = \eta\mu$. Indeed, it is sufficient and necessary to define $\mu(x) = (\rho(x), \rho(\overline{x})^T)$.

*Example 2 ([4]).* Let $M = \mathbb{B}^{n\times n}$. Then $M \times M^T = \mathbb{B}^{n\times n} \times (\mathbb{B}^{n\times n})^T$ is a submonoid of the set of $2n \times 2n$-Boolean matrices:

$$\mathbb{B}^{n\times n} \times \left(\mathbb{B}^{n\times n}\right)^T = \left\{ \begin{pmatrix} P & 0 \\ 0 & Q^T \end{pmatrix} \;\middle|\; P, Q \in \mathbb{B}^{n\times n} \right\} \text{ with } \overline{\begin{pmatrix} P & 0 \\ 0 & Q^T \end{pmatrix}} = \begin{pmatrix} Q & 0 \\ 0 & P^T \end{pmatrix}.$$

In the line above $P^T$ and $Q^T$ are the transposed matrices.

*Remark 3.* It is enough to prove Theorem 2 in the special case where $\mathbb{F} = A_\pm^*$ is a free monoid with involution and $\Phi$ is a conjunction of equations and constraints of the form $X \in \mu^{-1}(m)$, where $\mu : A_\pm^* \to N$ is a morphism between monoids with involution. Moreover, $A_\pm$ is a set where the involution has no fixed points.

### 3.5 The initial equation $W_{\text{init}}$

Recall that, due to Remark 3, we may assume that we are in the special situation where $\mathbb{F} = A_\pm^*$ and $A_\pm$ is a set where the involution has no fixed points. The next few steps are quite similar to those in the proof of Theorem 1. We introduce a special symbol # with $\overline{\#} = \#$, and we let $A = A_\pm \cup \{\#\}$ be the initial alphabet with involution. In this alphabet no other symbol except # is self-involuting.

All rational constraints are given by a morphism $\mu_{00} : A_\pm^* \to N$ where $N$ is a finite monoid with involution which has a zero. Without restriction we assume that $\mu_{00}(1)^{-1} \subseteq \{1\}$ and that for all $w \in A_\pm^*$ we have $\mu_{00}(w) \neq 0 \iff w \in \iota(A_\mathbb{F})$, where $\iota$ is the embedding of $A_\mathbb{F}$ into $A_\pm$ from above. We extend $\mu_{00}$ to a morphism $\mu_0 : A^* \to N$ by $\mu_0(\#) = 0$. We then extend $\mu_0$ to a morphism $\mu_{\text{init}} : (A \cup \Omega)^* \to N$ by guessing the values for $0 \neq \mu_{\text{init}}(X) = \overline{\mu_{\text{init}}(\overline{X})} \in N$. In the following $\mu_{\text{init}}(x)$ changes frequently for symbols outside of $A$, thus we use $\mu, \mu'$ as generic notation. However, the following will hold throughout: $\mu_0(a) = \mu_{\text{init}}(a) = \mu(a) = \mu'(a)$ for all $a \in A$. The starting point is now a system of equations $U_i = V_i$ with $U_i, V_i \in (A_\pm \cup \Omega)^*$ for $1 \leq i \leq s$ and the morphism $\mu_{\text{init}} : (A \cup \Omega)^* \to N$. We don't care that $|U_iV_i| = 3$ anymore.

We let $U' = U_1\# \cdots \#U_s$ and $V' = V_1\# \cdots \#V_s$. Therefore, we have only one single equation. Next, we define the actual initial equation $W_{\text{init}}$ as in (5).

$$W_{\text{init}} = \#x_1\# \ldots \#x_\ell\#U'\#V'\#\overline{U'}\#\overline{V'}\#\overline{x_\ell}\# \ldots \#\overline{x_1}\#. \tag{15}$$

In particular, according to (5) we have $A_\pm \cup \Omega = \{x_1, \ldots, x_\ell\}$ with $x_i = X_i$ for $1 \le i \le k$. The overall strategy to proving Theorem 2 is as before – we show that the following language is EDT0L:

$$\left\{\sigma(X_1)\#\cdots\#\sigma(X_k) \;\middle|\; \sigma(W_{\text{init}}) = \sigma(\overline{W_{\text{init}}}) \wedge \mu_{\text{init}} = \mu_0\sigma \wedge \forall X : \sigma(X) = \sigma(\overline{X})\right\}.$$

Recall that for every morphism $\sigma : \Omega \to A_\pm$ we have

$$\sigma(U') = \sigma(V') \iff \sigma(W_{\text{init}}) = \sigma(\overline{W_{\text{init}}}).$$

We fix some large enough $n = n_{\text{init}} \in \mathcal{O}(|W_{\text{init}}|)$ and alphabet $C$ of size $\mathcal{O}(n)$. We assume that $A \subseteq C$ and no other symbol than $\# \in C$ is self-involuting. In contrast to the above we content ourselves with $|C| \in \mathcal{O}(n)$; we leave it to the reader to calculate large enough constants.

As usual we let $A \subseteq B, B' \subseteq C$ and we assume that $B$ and $B'$ are closed under involution. By $\mathcal{X}$, $\mathcal{X}'$ we denote subsets of $\Omega$ which are closed under involution, too. Moreover, we let $\Sigma = C \cup \Omega$.

The letter $\mu$ refers to a morphism $\mu : (B \cup \mathcal{X})^* \to N$ defined by $\mu : B \cup \mathcal{X} \to N$ where $N$ is a finite monoid with involution. We assume that $\mu(a) = \mu_0(a)$ for all $a \in A$. The morphism $\mu$ encodes the rational constraints: in particular, using the appropriate $\mu$ we can guarantee that solutions are in $\iota(\mathbb{F})$.

### 3.6 Partial commutation induced by a type relation

The definition of type relations appearing in $\mathcal{G}$ was more general than necessary. For the proof of Theorem 2 we restrict the type relations in order to focus on what we need, because we use now compression arcs which have have not been used in Theorem 1. The main difference between the proofs of Theorem 1 and Theorem 2 is that reduced words may have self-involuting factors of the form $a\overline{a}$ and that compression of $a\overline{a}$ into a letter would lead to self-involuting letters, which we must avoid. The solution is simple: never compress any factor $a\overline{a}$. It will be enough to compress factors $(a\overline{a})^\ell$ for $\ell \ge 2$ down to a self-involuting word of length 2.

We restrict the rather general notion of *type relation* over $\Sigma = C \cup \Omega$ as follows. Since it is the restriction of the earlier definition, $\theta$ is an irreflexive and antisymmetric relation, where $(x, y) \in \theta$ implies $(\overline{x}, \overline{y}) \in \theta$. In addition, we require that first, $(x, y) \in \theta$ implies $y \in \{c, \overline{c}, c\overline{c}\}$ for some $c \in C$ and that there is no $(x', y') \in \theta$ with $x \in \{c, \overline{c}\}^*$, and second, only the following two forms of type relations $\theta$ are allowed, where $c \in C \setminus \{\#\}$ is a letter.

$$\theta \subseteq \{(x, c), (\overline{x}, \overline{c}) \mid x \in \Sigma \setminus \{c, \overline{c}\}\}. \tag{16}$$

$$\theta \subseteq \{(X, c\overline{c}) \mid X \in \Omega\} \cup \{(a\overline{a}, c\overline{c}) \mid a \in C \setminus \{c, \overline{c}\}\}. \tag{17}$$

Moreover, $|\theta(x)| \le 1$ where $\theta(x) = \{y \in C^+ \mid (x, y) \in \theta\}$. Clearly, we maintain $|\theta| \in \mathcal{O}(n)$, which allows us to store $\theta$ in quasi-linear space. Given $\theta$ and $\mu : B \cup \mathcal{X} \to N$ such that $\mu(xy) = \mu(yx)$ for all $(x, y) \in \theta$ we define as above the following two partially commutative monoids with involution.

1. $M(B,\mathcal{X},\theta,\mu) = (B\cup\mathcal{X})^*/\{xy = yx \mid (x,y)\in\theta\}$, a monoid with a morphism $\mu : M(B,\mathcal{X},\theta,\mu) \to N$.
2. $M(B,\theta,\mu) = B^*/\{xy = yx \mid (x,y)\in\theta\}$, a submonoid of $M(B,\mathcal{X},\theta,\mu)$ such that
$$\mu : M(B,\theta,\mu) \hookrightarrow M(B,\mathcal{X},\theta,\mu) \xrightarrow{\mu} N.$$

Let us recall that if $w \leq W \in M(B,\mathcal{X},\theta,\mu)$, then $w$ is called a *proper factor* if $w \neq 1$ and $|w|_{\#} = 0$ and that, since the defining relations for $M(B,\mathcal{X},\theta,\mu)$ are of the form $xy = xy$, we can define $|W|$ and $|W|_a$ for $W \in M(B,\mathcal{X},\theta,\mu)$ by representing $W$ by some word $W \in (B\cup\mathcal{X})^*$. It follows that we can decide $w \leq W$ in quasi-linear space for $w, W \in M(B,\mathcal{X},\theta,\mu)$.

As above, $W \in M(B,\mathcal{X},\theta,\mu)$ is *well-formed* if it is well-formed according to Definition 2. We repeat the definition of an extended equation and apply it to the restricted version of type relation.

**Definition 5.** *An* extended equation *is a tuple $V = (W,B,\mathcal{X},\theta,\mu)$ where $W \in M(B,\mathcal{X},\theta,\mu)$ is well-formed. A* $B$-solution *of $V$ is a $B$-morphism $\sigma : M(B\cup\mathcal{X},\theta,\mu) \to M(B,\theta,\mu)$ such that $\sigma(W) = \sigma(\overline{W})$ and $\sigma(X) \in c^*$ whenever $(X,c) \in \theta$. A* solution *of $V$ is a pair $(\alpha,\sigma)$ such that $\alpha : M(B,\theta,\mu) \to A^*$ is an $A$-morphism satisfying $\mu_0\alpha = \mu$ and $\sigma$ is a $B$-solution.*

### 3.7 The directed labeled graph $\mathcal{G}_{\mathbb{F}}$.

Define the graph $\mathcal{G}_{\mathbb{F}}$ to be the induced subgraph of $\mathcal{G}$ which is defined by the set of all extended equations $(W,B,\mathcal{X},\theta,\mu)$ where $\theta$ satisfies the specification as above. (If necessary, adopt the constant $\kappa$ to be large enough, say $\kappa = 100$.) In particular, $\theta$ is either of the form (16) or (17). The restriction is imposed in order to focus on the essential arcs, and it is allowed to start with $\mathcal{G}$ as defined originally. In particular, we keep the set of *initial vertices*, which are the vertices of the form
$$(W_{\text{init}}, A, \Omega, \emptyset, \mu_{\text{init}}).$$
The set of *final vertices* is again
$$\left\{(W,B,\emptyset,\emptyset,\mu) \;\middle|\; W = \overline{W}\right\}.$$

All arcs in $\mathcal{G}$ that are between vertices of $\mathcal{G}_{\mathbb{F}}$ are also arcs in $\mathcal{G}_{\mathbb{F}}$, since we consider the induced subgraph. In particular, Proposition 1 still holds, and states the following.

**Proposition 3.** *Let $V_0 \xrightarrow{h_1} V_1 \cdots \xrightarrow{h_t} V_t$ be a path in $\mathcal{G}_{\mathbb{F}}$ of length $t$, where $V_0 = (W_{\text{init}}, A, \Omega, \emptyset, \mu_{\text{init}})$ is an initial and $V_t = (W', B, \emptyset, \emptyset, \mu)$ is a final vertex. Then $V_0$ has a solution $(\mathrm{id}_A, \sigma)$ with $\sigma(W_{\text{init}}) = h_1\cdots h_t(W')$. Moreover, we have $W' \in \#u_1\#\cdots\#u_k\#B^*$ such that $|u_i|_{\#} = 0$ and we can write:*
$$h_1\cdots h_t(u_1\#\cdots\#u_k) = \sigma(X_1)\#\cdots\#\sigma(X_k), \tag{18}$$

*Proof.* See Proposition 1. □

## 4 General compression

We can now give the proof of Theorem 2, following the same scheme as for free groups and the proof of Theorem 1.

Consider an initial vertex $V_0 = (W_{\text{init}}, A, \Omega, \emptyset, \mu_{\text{init}})$ with a solution $(\alpha, \sigma)$. We show that $\mathcal{G}_{\mathbb{F}}$ contains a path $V_0 \xrightarrow{h_1} V_1 \cdots \xrightarrow{h_t} V_t$ to some final vertex $V_t = (W', B, \emptyset, \emptyset, \mu)$ such that $\sigma(W_{\text{init}}) = h_1 \cdots h_t(W')$. We show the existence of the path using a repetition of the sequence:

"block compression", "non-standard block compression", "pair compression".

Let us recall that the scheme is repeated until we reach a final vertex and that the procedures use some external knowledge about solutions. We proceed along arcs $V \xrightarrow{h} V'$ in $\mathcal{G}_{\mathbb{F}}$ thereby transforming a solution $(\alpha, \sigma)$ to $V$ into a solution $(\alpha', \sigma')$ to $V'$ such that we keep the invariant $\alpha\sigma(W) = \alpha' h \sigma'(W')$.

### 4.1 Standard block compression

The procedure has been described above in the main body of the paper as *block compression*. We start at a non-final vertex $V = (W, B, \mathcal{X}, \emptyset, \mu)$ with a solution $(\alpha, \sigma)$ with $|B| \leq |W| \in \mathcal{O}(n)$. We move along arcs satisfying the forward condition and we arrive at a vertex $V' = (W', B', \mathcal{X}', \emptyset, \mu')$ with a solution $(\alpha', \sigma')$. We have $|B|' \leq |W|' \in |W| + \mathcal{O}(n)$, so there is a possible increase by $\mathcal{O}(n)$ in the length of the equation, but we know that $W'$ does not contain any proper factor $b^2$ with $b \in B'$. At the end of the standard block compression we rename $V' = (W', B', \mathcal{X}', \emptyset, \mu')$ and $(\alpha', \sigma')$ as $V = (W, B, \mathcal{X}, \emptyset, \mu)$ $(\alpha, \sigma)$, but we keep in mind the increase of length by $\mathcal{O}(n)$.

After that, we start the non-standard block compression to remove all factors $a\overline{a}a$ from $\sigma(W)$. This is explained next.

### 4.2 Non-standard block compression

We follow the explanation and notation according to the main body of the paper. We consider some non-final vertex $V = (W, B, \mathcal{X}, \emptyset, \mu)$ with an empty type relation and a solution $(\alpha, \sigma)$. Let $B \setminus \{\#\} = B_+ \cup B_-$ be any partition such that $b \in B_+ \iff \overline{b} \in B_-$. Recall that $a\overline{a}$ could be a reduced word. As $a\overline{a}$ is a self-involuting word we cannot compress it into a single letter $c$ because then the letter $c$ is forced to be self-involuting, since compression of a factor $a\overline{a} = \overline{a\overline{a}}$ must be unambiguous. Note that $a\overline{a}$ has no non-trivial self-overlap since $a \neq \overline{a}$

Let $c \in C \setminus B$ and $h : C^* \to C^*$ be the renaming homomorphism defined by $h(c) = a$ and $h(\overline{c}) = \overline{a}$. Let $w \in B^*$ be any word. Then there is a unique word $w' \in B^*$ such that $w = h(w')$ and $w'$ does not have any factor $a\overline{a}$ though it may have a factor $\overline{a}a$. The word $w'$ can be obtained by replacing every occurrence of $a\overline{a}$ by $c\overline{c}$. The word $w'$ is unique because $a\overline{a}$ has no non-trivial self-overlap.

Let us introduce the following notation. For $a \in C$, $w \in \Sigma^* = (C \cup \Omega)^*$, and $\lambda \geq 1$ such that $(a\overline{a})^\lambda$ is a factor of $w$. We say that an occurrence of $(a\overline{a})^\lambda$ in

$w$ is *maximal* if the occurrence corresponds to a factorization $w = u(a\overline{a})^\lambda v$ such that neither $u \in \Sigma^* a\overline{a}$ nor $v \in a\overline{a}\Sigma^*$. This means that at least one occurrence of $(a\overline{a})^\lambda$ in $w$ is not contained in any occurrence of a factor $(a\overline{a})^{\lambda+1}$. Note that $(a\overline{a})^1$, $(a\overline{a})^2$, $(a\overline{a})^3$ may be factors in some $w$, both $(a\overline{a})^1$ and $(a\overline{a})^2$ have maximal occurrences, but $(a\overline{a})^3$ does not: for example, $w = (a\overline{a})^1\#(a\overline{a})^2\#(a\overline{a})^4$. Note also that $a\overline{a}$ has a maximal occurrence in $w = \overline{a}a\overline{a}a$.

We can repeat, partly verbatim, the standard block compression, but there are slight modifications, and we divide the procedure into smaller steps.

*Remark 4.* Before we describe the procedure in mathematical terms, let us try to give a a high level explanation what a non-standard block compression does. The basic idea is simple. In order to avoid self-involuting letters we cannot compress $c\overline{c}$ into a single letter, but we can compress $c\overline{c}c\overline{c}$ into the word $c\overline{c}$. This means we can compress maximal blocks $(c\overline{c})^{2\ell}$ into blocks $(c\overline{c})^\ell$. The compression must correspond to morphisms. This is fine: the morphism $c \mapsto c\overline{c}$ maps $c\overline{c}$ to $c\overline{c}c\overline{c}$. But then we can continue the same way only if $\ell$ is even. Therefore there is some extra work necessary if $\ell$ becomes odd. If there is a maximal block $(c\overline{c})^\ell$ with $\ell$ odd then we replace first $(c\overline{c})^\ell$ by $c_\lambda\overline{c_\lambda}(c\overline{c})^{\ell-1}$ where $c_\lambda$ is a fresh letter. Once we have $c_\lambda\overline{c_\lambda}$ available, we can compress $(c\overline{c})c_\lambda\overline{c_\lambda}(c\overline{c})$ into $c_\lambda\overline{c_\lambda}$. The morphism maps $c_\lambda$ to $c\overline{c}c_\lambda$. Thus, $c_\lambda\overline{c_\lambda}$ is mapped to $(c\overline{c})c_\lambda\overline{c_\lambda}(c\overline{c})$ which is equal to $c_\lambda\overline{c_\lambda}(c\overline{c})^2$ due to partial commutation. So at the end every maximal block $(c\overline{c})^\ell$ where $\ell$ is even or odd gets compressed into some $c_\lambda\overline{c_\lambda}$. One could say that $c_\lambda\overline{c_\lambda}$ is a special pair with a sort of marker which always guesses correctly whether it sits "inside" some block $(c\overline{c})^\ell$ with $\ell \equiv 2 \bmod 4$ or with $\ell \equiv 0 \bmod 4$. This is why counting mod4 comes in.

**begin non-standard block compression**

We begin at vertex $V = (W, B, \mathcal{X}, \emptyset, \mu)$ with a solution $(\alpha, \sigma)$. During the process we introduce partial commutation but it vanishes at the end. After each transformation we rename the current vertex as $(W, B, \mathcal{X}, \theta, \mu)$ with a solution $(\alpha, \sigma)$. During the $(a\overline{a})$-compression we enlarge the alphabet and then the standard notation for a vertex becomes $(W, B', \mathcal{X}, \theta, \mu)$.

1. Follow arcs of type **4** and **6** in order to remove all variables with $|\sigma(X)| \leq 10$. Thus, without restriction, we have $|\sigma(X)| > 10$ for all $X$. If $V$ became final, we are done and we stop.
2. For each $X$ we now have $\sigma(X) = bw$ for some $b \in B$ and $w \in B^+$. Following a substitution arc **6**, we replace $X$ by $bX$, $\overline{X}$ by $\overline{X}\,\overline{b}$ and change $\mu(X)$ to $\mu(X) = \overline{\mu(\overline{X})} = \mu(w)$. Now, if $bX \leq W$ and $b'X \leq W$ are factors with $b, b' \in B$ then $\# \neq b = b'$ due to the previous substitution $X \mapsto bX$.
3. For each $a \in B_+$ define sets $\Lambda_a \subseteq \mathbb{N}$ which contain those $\lambda \geq 1$ such that there is a maximal occurrence of $(a\overline{a})^\lambda$ in $\sigma(W)$ where at least one of the $a$'s is visible. Note that we treat $a \in B_+$ different from $\overline{a} \in B_-$. However, this is not essential here. If there is a maximal occurrence of $(a\overline{a})^\lambda$ in $\sigma(W)$ where at least one of the $a$'s is visible then there is another maximal occurrence of $(a\overline{a})^\lambda$ in $\sigma(W)$ where at least one of the $\overline{a}$'s is visible. We have $\sum_{a\in B_+} |\Lambda_a| \leq$

$|W|$. We also let

$$\mathcal{X}_a = \{X \in \mathcal{X} \mid aX \leq W \wedge \sigma(X) \in \overline{a}B^* \vee \overline{a}X \leq W \wedge \sigma(X) \in a\overline{a}B^*\}\,.$$

Note that if $X \in \mathcal{X}_a$ for some $a \in B_+$ then either the factor $a\overline{a}$ or the factor $\overline{a}a$ or both have a "crossing" in $\sigma(W)$.

4. For each $\Lambda_a \neq \emptyset$ – one after another – run the following subroutine, called $(a\overline{a})$-*compression*. The purpose is to remove all proper factors $(a\overline{a})^\ell$ with $a \in B_+$ and $\ell \geq 1$ from $W$. More precisely, if $(a\overline{a})^\ell$ is a maximal occurrence of that factor in $W$ then the following $(a\overline{a})$-compression replaces this occurrence $(a\overline{a})^\ell$ by some factor $c_\lambda\overline{c_\lambda}$. We do not have $\ell = \lambda$ in general, but clearly $\left|(a\overline{a})^\ell\right| \geq |c_\lambda\overline{c_\lambda}| = 2$.

**end non-standard block compression**

**Subroutine $(a\overline{a})$-compression.**

The subroutine is called at a vertex $V = (W, B, \mathcal{X}, \emptyset, \mu)$ with solution $(\alpha, \sigma)$.
**begin** $(a\overline{a})$**-compression**

1. Introduce fresh letters $c_a, \overline{c_a}$ with $\mu(c_a) = \mu(a)$. In addition, for each $\lambda \in \Lambda_a$ introduce fresh letters $c_{\lambda,a}, \overline{c_{\lambda,a}}$ with $\mu(c_{\lambda,a}) = \mu(a)$. Define $h(c_{\lambda,a}) = h(c_a) = a$ and introduce a type by letting

$$\theta = \{(c_{\lambda,a}\overline{c_{\lambda,a}}, c_a\overline{c_a}) \mid \lambda \in \Lambda_a\}\,.$$

Renaming arcs **1** realizes this transformation. We did not touch $W$, hence $W = h(W)$, but we enlarged $B$ to some set $B'$ and we introduced partial commutation. We abbreviate $c = c_a$, $\overline{c} = \overline{c_a}$, $c_\lambda = c_{\lambda,a}$, and $\overline{c_\lambda} = \overline{c_{\lambda,a}}$.

2. (Change $W$ and its solution.) We replace in $\sigma(W) \in B^*$ every maximal occurrence of a factor $(a\overline{a})^\lambda$ with $\lambda \in \Lambda_a$ by $(c\overline{c})^\lambda$. This yields a new word $W' \in B'^*$. The transformation can be realized again by a single renaming arc defined by $h(c) = a$ and leading to a solution $(\alpha', \sigma')$.
Various $c$ and $\overline{c}$ appear in $W'$ and $\sigma'(W')$. Note that every $c$ in $\sigma'(W')$ is followed by some $\overline{c}$ and every $\overline{c}$ is preceded by some $c$. Some of their positions are visible. For example, $W'$ may have factors of the form $X\overline{c}Y$ or $X\overline{c}c\overline{c}cY$ etc. We rename the vertex and its solution as $(W, B', \mathcal{X}, \theta, \mu)$ and $(\alpha, \sigma)$.
3. Consider all $X \in \mathcal{X}_a$ where $\sigma(X)$ is a factor of some word in $(c\overline{c})^*$. (For example, $\sigma(X) \in \overline{c}(c\overline{c})^*c$.) Follow substitution arcs **6** such that first, for the resulting solution $\sigma'$ we have $\sigma'(X) \in (c\overline{c})^{4m}$ for some $m \in \mathbb{N}$ and second, each occurrence of such $X \in \mathcal{X}_a$ in $W'$ occurs inside an occurrence of $c\overline{c}X$. As usual, we rename the vertex and its solution as $(W, B', \mathcal{X}, \theta, \mu)$ and $(\alpha, \sigma)$.
4. (Typing variables.) Enlarge $\theta$ such that it becomes

$$\theta = \{(c_\lambda\overline{c_\lambda}, c\overline{c}) \mid \lambda \in \Lambda_a\} \cup \{(X, c\overline{c}) \mid X \in \mathcal{X}_a \wedge \sigma(X) \in (c\overline{c})^*\}\,.$$

5. For all $X \in \mathcal{X}_a$ where $\sigma(X) \notin (c\overline{c})^*$ factorize $\sigma(X) = uv$ such that $u$ is a suffix (possibly empty) of some word in $(c\overline{c})^*$ and $v \notin c\overline{c}M(B', \theta, \mu)$. Following more substitution arcs we can make sure that first, $u \in (c\overline{c})^{4m}$ for some $m \in \mathbb{N}$ and second, every occurrence of such $X$ is $W$ occurs as a factor $c\overline{c}X$. This is due to the definition of $\mathcal{X}_a$.
   After renaming we have either $\sigma(X) \in ((c\overline{c})^4)^*$ or $\sigma(X) \in ((c\overline{c})^4)^* v ((c\overline{c})^4)^*$ with $1 \neq v \notin c\overline{c}M(B', \theta, \mu) \cup M(B', \theta, \mu)c\overline{c}$.
6. Remove all variables with $\sigma(X) = 1$ and rename the current vertex as $(W, B', \mathcal{X}, \theta, \mu)$.
7. Call the following while loop, called *$\Lambda_a$-compression*. Repeat the loop until $\Lambda_a = \{0\}$. In the beginning of the loop we have $\Lambda_a \neq \emptyset$ and we don't have $0 \in \Lambda_a$, but the number 0 sneaks in. Actually we will have that $\Lambda_a = \{0\}$ if and only if and there are no more factors $c\overline{c}$. Note that there at least as many subsets $\{c_\lambda, \overline{c_\lambda}\}$ available as there are numbers in $\Lambda_a$. During the following process we will mark some $c_\lambda$, $\overline{c_\lambda}$ to make sure that we use each set $\{c_\lambda, \overline{c_\lambda}\}$ only once. At the beginning all $c_\lambda$ are unmarked.

   **begin $\Lambda_a$-compression**
   As above, we realize each transformation via arcs satisfying the forward property. We continue to denote, by default, each current vertex as $(W, B', \mathcal{X}, \theta, \mu)$ and the current solution as $(\alpha, \sigma)$.
   The first while loop is realized as a path in $\mathcal{G}_{\mathbb{F}}$ by renaming arcs with label $h(c_\lambda) = c$; the second one uses substitution arcs with label $h(c_\lambda) = c\overline{c}c_\lambda$.

   Let $\Lambda = \Lambda_a$.
   **while** $\Lambda \neq \{0\}$ **do**
   (a) **while** there is $\ell \in \Lambda$ such that $\ell$ is odd **do**
      – let $\ell$ be largest odd number in $\Lambda$;
      – choose an unmarked letter $c_\lambda$ and mark $c_\lambda$ and $\overline{c_\lambda}$;
      – replace every maximal occurrence of $(c\overline{c})^\ell$ by $c_\lambda\overline{c_\lambda}(c\overline{c})^{\ell-1}$;
      whenever possible make the factor $c_\lambda\overline{c_\lambda}$ visible;
      (we have $c_\lambda\overline{c_\lambda}(c\overline{c})^{\ell-1} = (c\overline{c})^{\ell_1}c_\lambda\overline{c_\lambda}(c\overline{c})^{\ell_2} \in M(B', \theta, \mu)$ for all $\ell_1 + \ell_2 = \ell - 1$, hence $c_\lambda\overline{c_\lambda}$ can be chosen to be visible for an occurrence of $(c\overline{c})^\ell$ unless no position of that occurrence is visible)
      – replace $\Lambda$ by $(\Lambda \setminus \{\ell\}) \cup \{\ell - 1\}$.
   **end while**
   Note that now, if $(c\overline{c})^\ell$ is a maximal occurrence in $\sigma(W)$ then $\ell \in \Lambda$ and all $\ell \in \Lambda$ are even.
   (b) **while** $\sigma(W)$ has a maximal occurrence of some factor $c_\lambda\overline{c_\lambda}(c\overline{c})^\ell$ with $\ell \equiv 2 \bmod 4$ **do**
      – let $\ell$ be largest number in $\Lambda$ such that there is a maximal occurrence of the factor $c_\lambda\overline{c_\lambda}(c\overline{c})^\ell$ with $\ell \equiv 2 \bmod 4$;
      – replace every occurrence of $c_\lambda\overline{c_\lambda}(c\overline{c})^\ell$ by $c_\lambda\overline{c_\lambda}(c\overline{c})^{\ell-2}$;
      realize this transformation via some substitution arc with label $h(c_\lambda) = c\overline{c}c_\lambda$; note that $h(c_\lambda\overline{c_\lambda}) = c\overline{c}c_\lambda\overline{c_\lambda}c\overline{c} = c_\lambda\overline{c_\lambda}(c\overline{c})^2) \in M(B', \theta, \mu)$
      – replace $\Lambda$ by $\Lambda \cup \{\ell - 2\}$;

– if $\sigma(W)$ does not contain any maximal occurrence of $(c\overline{c})^\ell$ then replace $\Lambda$ by $\Lambda \setminus \{\ell\}$.

**end while**

Note that the first two while loops did not use any substitution arc since no "uncrossing" was necessary. Moreover, now all $\ell \in \Lambda$ are even and if there is a maximal occurrence of some factor $c_\lambda\overline{c_\lambda}(c\overline{c})^\ell$ then we have $\ell \equiv 0 \bmod 4$. Thus dividing such value $\ell$ by two keeps this value even.

(c) Follow a substitution arc with label $h(c) = c\overline{c}$ in order to replace all maximal occurrences of factors $(c\overline{c})^\ell$ with $\ell \in \Lambda$ by $(c\overline{c})^{\ell/2}$;
replace $\Lambda$ by the set $\{\lambda/2 \mid \lambda \in \Lambda\}$;
(note that $\Lambda$ may have odd numbers, again)

(d) Remove all variables $X$ with $|\sigma(X)| \leq 10$.

(e) For each of the remaining $X \in \mathcal{X}_a$ there is a unique factorization $\sigma(X) = uv$ such that $u \in (c\overline{c})^*$ and $v \notin (c\overline{c})^* M(B', \theta, \mu)$. Using substitution arcs we may assume that $u \in (c\overline{c})^{4m}$ for some $m \in \mathbb{N}$.

**end while**

**end $\Lambda_a$-compression**

8. Let $B = B' \setminus \{c, \overline{c}\}$. Since $\Lambda_a = \{0\}$ after at the end of the $\Lambda_a$-compression, no $c$ or $\overline{c}$ appears in $\sigma(W)$: they are all compressed into single letters $c_\lambda$ or $\overline{c_\lambda}$. Moreover, for $x \in B \cup \mathcal{X}$ we have $\theta(x) = \emptyset$. Hence we can follow an alphabet reduction arc $(W, B', \mathcal{X}, \theta, \mu) \xrightarrow{\varepsilon} (W, B, \mathcal{X}, \emptyset, \mu)$. The new solution to $(W, B, \mathcal{X}, \emptyset, \mu)$ is the pair $(\alpha', \sigma)$ where $\alpha' = \alpha\varepsilon$ is defined by the restriction of $\alpha$ to the free monoid $M(B, \emptyset, \mu')$. We rename the current vertex and its solution as $(W, B, \mathcal{X}, \emptyset, \mu)$ and $(\alpha, \sigma)$.

**end $(a\overline{a})$-compression**

Having performed one round of a non-standard block compressions, we have increased the length of $W$ by at most $\mathcal{O}(n)$. We end up at a vertex named again as $V = (W, B, \mathcal{X}, \emptyset, \mu)$ which has a solution $(\alpha, \sigma)$. The difference is that $W$ has no proper factor $a\overline{a}a$ with $a \in B$ anymore.

### Repeat: 1. standard block compression, 2. non-standard block compression, 3. pair compression until $\mathcal{X} = \emptyset$

The title of this section explains what we do. We repeat rounds of 1. standard block compression, 2. non-standard block compression, 3. pair compression. After the non-standard block compression the word $W$ does not contain any proper factor of the form $a^\ell$ with $\ell \geq 2$ or $a\overline{a}a$ where $a \in B$. Thus, at the beginning of pair compression a proper factor $u$ of $W$ of length three looks as $u = abc$ with either $c \notin \{b, \overline{b}\}$ or $a \notin \{b, \overline{b}\}$.

To see that, indeed, all factors $a\overline{a}a$ vanish, consider for example the case that before (and after) the standard block compression the word $W$ contains a factor $ba\overline{a}ab$ with $a \neq b$. After that the non-standard block compression uses renaming. It changes this factor either to $bc\overline{c}ab$ (if $a \in B_+$) or to $ba\overline{c}cb$ (if $\overline{a} \in B_+$); and

at the end of the non-standard block compression this factor appears either as $bc_1\overline{c}_1ab$ or as $ba\overline{c}_1c_1b$.

The pair compression we proceed according to Section 2.9, but we never compress any pair $a\overline{a}$. Now consider again $u \leq W$ with $u = abc$ and either $c \notin \{b, \overline{b}\}$ or $a \notin \{b, \overline{b}\}$. By symmetry, we may assume $a \notin \{b, \overline{b}\}$. Now the probability that $ab$ is compressed is $\Pr[\, a \in L \wedge b \in R\,] = \Pr[\, a \in L\,] \cdot \Pr[\, b \in R\,] = \frac{1}{4}$. It follows that the expected length of that factor $u$ after pair compression is at most $3 \cdot \frac{3}{4} + 2 \cdot \frac{1}{4} = \frac{11}{4}$ which is less than 3. Thus, we obtain a recursion of type $s(1) \in \mathcal{O}(n)$ and $s(i+1) \leq q \cdot s(i) + \mathcal{O}(n)$ for all $i \in \mathbb{N}$ where $q = \frac{11}{12} < 1$. Such a recursion implies $s(i) \in \mathcal{O}(n)$ for all $i \in \mathbb{N}$. This proves Theorem 2. □

## References


[1] A. V. Aho. Indexed grammars—an extension of context-free grammars. *J. Assoc. Comput. Mach.*, 15:647–671, 1968.

[2] P. R. Asveld. Controlled iteration grammars and full hyper-AFL's. *Information and Control*, 34(3):248 – 269, 1977.

[3] M. Benois. Parties rationelles du groupe libre. *C. R. Acad. Sci. Paris, Sér. A*, 269:1188–1190, 1969.

[4] V. Diekert, C. Gutiérrez, and Ch. Hagenah. The existential theory of equations with rational constraints in free groups is PSPACE-complete. *Information and Computation*, 202:105–140, 2005. Conference version in STACS 2001, LNCS 2010, 170–182, 2004.

[5] V. Diekert, A. Jeż, and W. Plandowski. Finding all solutions of equations in free groups and monoids with involution. In E. A. Hirsch, S. O. Kuznetsov, J. Pin, and N. K. Vereshchagin, editors, *Computer Science Symposium in Russia 2014, CSR 2014, Moscow, Russia, June 7-11, 2014. Proceedings*, volume 8476 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2014.

[6] A. Ehrenfeucht and G. Rozenberg. On some context free languages that are not deterministic ET0L languages. *RAIRO Theor. Inform. Appl.*, 11:273–291, 1977.

[7] S. Eilenberg. *Automata, Languages, and Machines*, volume A. Academic Press, New York and London, 1974.

[8] J. Ferté, N. Marin, and G. Sénizergues. Word-mappings of level 2. *Theory Comput. Syst.*, 54:111–148, 2014.

[9] R. H. Gilman. Personal communication, 2012.

[10] S. Jain, A. Miasnikov, and F. Stephan. The complexity of verbal languages over groups. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25-28, 2012*, pages 405–414. IEEE Computer Society, 2012.

[11] A. Jeż. Recompression: a simple and powerful technique for word equations. In N. Portier and T. Wilke, editors, *STACS*, volume 20 of *LIPIcs*, pages 233–244, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. To appear in JACM.

[12] O. Kharlampovich and A. Myasnikov. Elementary theory of free non-abelian groups. *J. of Algebra*, 302:451–552, 2006.

[13] Ch. H. Papadimitriou. *Computational Complexity*. Addison Wesley, 1994.

[14] W. Plandowski. An efficient algorithm for solving word equations. In J. M. Kleinberg, editor, *STOC*, pages 467–476. ACM, 2006.

[15] W. Plandowski. Personal communication, 2014.

[16] W. Plandowski and W. Rytter. Application of Lempel-Ziv encodings to the solution of word equations. In K. G. Larsen et al., editors, *Proc. 25th International Colloquium Automata, Languages and Programming (ICALP'98), Aalborg (Denmark), 1998*, volume 1443 of *Lecture Notes in Computer Science*, pages 731–742, Heidelberg, 1998. Springer-Verlag.

[17] A. A. Razborov. *On Systems of Equations in Free Groups.* PhD thesis, Steklov Institute of Mathematics, 1987. In Russian.

[18] A. A. Razborov. On systems of equations in free groups. In *Combinatorial and Geometric Group Theory*, pages 269–283. Cambridge University Press, 1994.

[19] G. Rozenberg and A. Salomaa. *The Book of L.* Springer, 1986.

[20] G. Rozenberg and A. Salomaa, editors. *Handbook of Formal Languages*, volume 1. Springer, 1997.

[21] Z. Sela. Diophantine geometry over groups VIII: Stability. *Annals of Math.*, 177:787–868, 2013.